

Studies in Computational Intelligence 1142

Hocine Cherifi
Luis M. Rocha
Chantal Cherifi
Murat Donduran *Editors*

Complex Networks & Their Applications XII

Proceedings of The Twelfth
International Conference on Complex
Networks and their Applications:
COMPLEX NETWORKS 2023, Volume 2

 Springer

Series Editor

Janusz Kacprzyk, *Polish Academy of Sciences, Warsaw, Poland*

The series “Studies in Computational Intelligence” (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

Indexed by SCOPUS, DBLP, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

Hocine Cherifi · Luis M. Rocha ·
Chantal Cherifi · Murat Donduran
Editors

Complex Networks & Their Applications XII

Proceedings of The Twelfth International
Conference on Complex Networks and their
Applications: COMPLEX NETWORKS 2023,
Volume 2

Editors

Hocine Cherifi 
University of Burgundy
Dijon Cedex, France

Chantal Cherifi
IUT Lumière - Université Lyon 2
University of Lyon
Bron, France

Luis M. Rocha
Thomas J. Watson College of Engineering
and Applied Science
Binghamton University
Binghamton, NY, USA

Murat Donduran
Department of Economics
Yildiz Technical University
Istanbul, Türkiye

ISSN 1860-949X

ISSN 1860-9503 (electronic)

Studies in Computational Intelligence

ISBN 978-3-031-53498-0

ISBN 978-3-031-53499-7 (eBook)

<https://doi.org/10.1007/978-3-031-53499-7>

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2024

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

Preface

Dear Colleagues, Participants, and Readers,

We present the 12th Complex Networks Conference proceedings with great pleasure and enthusiasm. Like its predecessors, this edition proves complex network research's ever-growing significance and interdisciplinary nature. As we navigate the intricate web of connections that define our world, understanding complex systems, their emergent properties, and the underlying structures that govern them has become increasingly crucial.

The Complex Networks Conference has established itself as a pivotal platform for researchers, scholars, and experts from various fields to converge, exchange ideas, and push the boundaries of knowledge in this captivating domain. Over the past twelve years, we have witnessed remarkable progress, breakthroughs, and paradigm shifts highlighting the dynamic and complex tapestry of networks surrounding us, from biological systems and social interactions to technological infrastructures and economic networks.

This year's conference brought together an exceptional cohort of experts, including our keynote speakers:

- Michael Bronstein, University of Oxford, UK, enlightened us on “Physics-inspired Graph Neural Networks”
- Kathleen Carley, Carnegie Mellon University, USA, explored “Coupling in High Dimensional Networks”
- Manlio De Domenico, University of Padua, Italy, introduced “An Emerging Framework for the Functional Analysis of Complex Interconnected Systems”
- Danai Koutra, University of Michigan, USA, shared insights on “Advances in Graph Neural Networks: Heterophily and Beyond”
- Romualdo Pastor-Satorras, UPC, Spain, discussed “Opinion Depolarization in Interdependent Topics and the Effects of Heterogeneous Social Interactions”
- Tao Zhou, USTC, China, engaged us in “Recent Debates in Link Prediction”

These renowned experts addressed a spectrum of critical topics and the latest methodological advances, underscoring the continued expansion of this field into ever more domains.

We were also fortunate to benefit from the expertise of our tutorial speakers on November 27, 2023:

- Tiago de Paula Peixoto, CEU Vienna, Austria, guided “Network Inference and Reconstruction”
- Maria Liakata, Queen Mary University of London, UK, led us through “Longitudinal language processing from user-generated content”

We want to express our deepest gratitude to all the authors, presenters, reviewers, and attendees who have dedicated their time, expertise, and enthusiasm to make this event successful. The peer-review process, a cornerstone of scientific quality, ensures

that the papers in these proceedings have undergone rigorous evaluation, resulting in high-quality contributions.

We encourage you to explore the rich tapestry of knowledge and ideas as we dive into these four proceedings volumes. The papers presented here represent not only the diverse areas of research but also the collaborative and interdisciplinary spirit that defines the complex networks community.

In closing, we extend our heartfelt thanks to the organizing committees and volunteers who have worked tirelessly to make this conference a reality. We hope these proceedings inspire future research, innovation, and collaboration, ultimately helping us better understand the world's networks and their profound impacts on science, technology, and society.

We hope that the pleasure you have reading these papers matches our enthusiasm for organizing the conference and assembling this collection of articles.

Hocine Cherifi
Luis M. Rocha
Chantal Cherifi
Murat Donduran

Organization and Committees

General Chairs

Hocine Cherifi
Luis M. Rocha

University of Burgundy, France
Binghamton University, USA

Advisory Board

Jon Crowcroft
Raissa D'Souza
Eugene Stanley
Ben Y. Zhao

University of Cambridge, UK
Univ. of California, Davis, USA
Boston University, USA
University of Chicago, USA

Program Chairs

Chantal Cherifi
Murat Donduran

University of Lyon, France
Yildiz Technical University, Turkey

Lightning Chairs

Konstantin Avrachenkov
Mathieu Desroches
Huijuan Wang

Inria Université Côte d'Azur, France
Inria Université Côte d'Azur, France
TU Delft, Netherlands

Poster Chairs

Christophe Crespelle
Manuel Marques Pita
Laura Ricci

Université Côte d'Azur, France
Universidade Lusófona, Portugal
University of Pisa, Italy

Special Issues Chair

Sabrina Gaito University of Milan, Italy

Publicity Chairs

Fabian Braesemann University of Oxford, UK
Zachary Neal Michigan State University, USA
Xiangjie Kong Dalian University of Technology, China

Tutorial Chairs

Luca Maria Aiello Nokia-Bell Labs, UK
Leto Peel Maastricht University, Netherlands

Social Media Chair

Brennan Klein Northeastern University, USA

Sponsor Chairs

Roberto Interdonato CIRAD - UMR TETIS, France
Christophe Cruz University of Burgundy, France

Sustainability Chair

Madeleine Aurelle City School International De Ferney-Voltaire,
France

Local Committee Chair

Charlie Joyez Université Côte d'Azur, France

Giacomo Baggio	University of Padova, Italy
Franco Bagnoli	Università di Firenze, Italy
James Bagrow	University of Vermont, USA
Yiguang Bai	Xidian University, China
Sven Banisch	Karlsruhe Institute of Technology, Germany
Annalisa Barla	Università degli Studi di Genova, Italy
Nikita Basov	The University of Manchester, UK
Anais Baudot	CNRS, AMU, France
Gareth J. Baxter	University of Aveiro, Portugal
Loredana Bellantuono	University of Bari Aldo Moro, Italy
Andras Benczur	SZTAKI, Hungary
Rosa M. Benito	Universidad Politécnica de Madrid, Spain
Ginestra Bianconi	Queen Mary University of London, UK
Ofer Biham	The Hebrew University, Israel
Romain Billot	IMT Atlantique, France
Livio Bioglio	University of Turin, Italy
Hanjo D. Boekhout	Leiden University, Netherlands
Anthony Bonato	Toronto Metropolitan University, Canada
Anton Borg	Blekinge Institute of Technology, Sweden
Cecile Bothorel	IMT Atlantique, France
Federico Botta	University of Exeter, UK
Romain Bourqui	University of Bordeaux, France
Alexandre Bovet	University of Zurich, Switzerland
Dan Braha	New England Complex Systems Institute, USA
Ulrik Brandes	ETH Zürich, Switzerland
Rion Brattig Correia	Instituto Gulbenkian de Ciência, Portugal
Chico Camargo	University of Exeter, UK
Gian Maria Campedelli	Fondazione Bruno Kessler, Italy
M. Abdullah Canbaz	University at Albany SUNY, USA
Vincenza Carchiolo	DIEEI, Italy
Dino Carpentras	ETH Zürich, Switzerland
Giona Casiraghi	ETH Zürich, Switzerland
Douglas Castilho	Federal Inst. of South of Minas Gerais, Brazil
Costanza Catalano	University of Florence, Italy
Lucia Cavallaro	Free University of Bozen/Bolzano, Italy
Remy Cazabet	University of Lyon, France
Jianrui Chen	Shaanxi Normal University, China
Po-An Chen	National Yang Ming Chiao Tung Univ., Taiwan
Xihui Chen	University of Luxembourg, Luxembourg
Sang Chin	Boston University, USA
Daniela Cialfi	Institute for Complex Systems, Italy
Giulio Cimini	University of Rome Tor Vergata, Italy

Matteo Cinelli	Sapienza University of Rome, Italy
Salvatore Citraro	University of Pisa, Italy
Jonathan Clarke	Imperial College London, UK
Richard Clegg	QMUL, UK
Reuven Cohen	Bar-Ilan University, Israel
Jean-Paul Comet	Université Côte d'Azur, France
Marco Coraggio	Scuola Superiore Meridionale, Italy
Michele Coscia	ITU Copenhagen, Denmark
Christophe Crespelle	Université Côte d'Azur, France
Regino H. Criado Herrero	Universidad Rey Juan Carlos, Spain
Marcelo V. Cunha	Instituto Federal da Bahia, Brazil
David Soriano-Paños	Instituto Gulbenkian de Ciência, Portugal
Joern Davidsen	University of Calgary, Canada
Toby Davies	University of Leeds, UK
Caterina De Bacco	Max Planck Inst. for Intelligent Systems, Germany
Pietro De Lellis	University of Naples Federico II, Italy
Pasquale De Meo	University of Messina, Italy
Domenico De Stefano	University of Trieste, Italy
Fabrizio De Vico Fallani	Inria-ICM, France
Charo I. del Genio	Coventry University, UK
Robin Delabays	HES-SO, Switzerland
Yong Deng	Univ. of Electronic Science and Tech., China
Mathieu Desroches	Inria Centre at Université Côte d'Azur, France
Carl P. Dettmann	University of Bristol, UK
Zengru Di	Beijing Normal University, China
Riccardo Di Clemente	Northeastern University London, UK
Branco Di Fátima	University of Beira Interior (UBI), Portugal
Alessandro Di Stefano	Teesside University, UK
Ming Dong	Central China Normal University, China
Constantine Dovrolis	Georgia Tech, USA
Maximilien Dreveton	EPFL, Switzerland
Ahlem Drif	University of Setif, Algeria
Johan L. Dubbeldam	Delft University of Technology, Netherlands
Jordi Duch	Universitat Rovira i Virgili, Spain
Cesar Ducruet	CNRS, France
Mohammed El Hassouni	Mohammed V University in Rabat, Morocco
Frank Emmert-Streib	Tampere University, Finland
Gunes Ercal	Southern Illinois University Edwardsville, USA
Alejandro Espinosa-Rada	ETH Zürich, Switzerland
Alexandre Evsukoff	Universidade Federal do Rio de Janeiro, Brazil
Mauro Faccin	University of Bologna, Italy

Max Falkenberg	City University, UK
Guilherme Ferraz de Arruda	CENTAI Institute, Italy
Andrea Flori	Politecnico di Milano, Italy
Manuel Foerster	Bielefeld University, Germany
Emma Fraxanet Morales	Pompeu Fabra University, Spain
Angelo Furno	LICIT-ECO7, France
Sergio Gómez	Universitat Rovira i Virgili, Spain
Sabrina Gaito	Università degli Studi di Milano, Italy
José Manuel Galán	Universidad de Burgos, Spain
Alessandro Galeazzi	Ca' Foscari university of Venice, Italy
Lazaros K. Gallos	Rutgers University, USA
Joao Gama	INESC TEC—LIAAD, Portugal
Jianxi Gao	Rensselaer Polytechnic Institute, USA
David Garcia	University of Konstanz, Germany
Floriana Gargiulo	CNRS, France
Michael T. Gastner	Singapore Institute of Technology, Singapore
Alexander Gates	University of Virginia, USA
Alexandra M. Gerbasi	Exeter Business School, UK
Fakhteh Ghanbarnejad	Potsdam Inst. for Climate Impact Res., Germany
Cheol-Min Ghim	Ulsan National Inst. of Science and Tech., South Korea
Tommaso Gili	IMT School for Advanced Studies Lucca, Italy
Silvia Giordano	Univ. of Applied Sciences of Southern Switzerland, Switzerland
Rosalba Giugno	University of Verona, Italy
Kimberly Glass	Brigham and Women's Hospital, USA
David Gleich	Purdue University, USA
Antonia Godoy Lorite	UCL, UK
Kwang-Il Goh	Korea University, South Korea
Carlos Gracia	University of Zaragoza, Spain
Oscar M. Granados	Universidad Jorge Tadeo Lozano, Colombia
Michel Grossetti	CNRS, France
Guillaume Guerard	ESILV, France
Jean-Loup Guillaume	Université de la Rochelle, France
Furkan Gursoy	Bogazici University, Turkey
Philipp Hövel	Saarland University, Germany
Meesoon Ha	Chosun University, South Korea
Bianca H. Habermann	AMU, CNRS, IBDM UMR 7288, France
Chris Hankin	Imperial College London, UK
Yukio Hayashi	JAIST, Japan
Marina Hennig	Johannes Gutenberg University of Mainz, Germany

Takayuki Hiraoka	Aalto University, Finland
Marion Hoffman	Institute for Advanced Study in Toulouse, France
Bernie Hogan	University of Oxford, UK
Seok-Hee Hong	University of Sydney, Australia
Yujie Hu	University of Florida, USA
Flavio Iannelli	UZH, Switzerland
Yuichi Ikeda	Kyoto University, Japan
Roberto Interdonato	CIRAD, France
Antonio Iovanella	Univ. degli Studi Internazionali di Roma, Italy
Arkadiusz Jędrzejewski	CY Cergy Paris Université, France
Tao Jia	Southwest University, China
Jiaojiao Jiang	UNSW Sydney, Australia
Di Jin	University of Michigan, USA
Ivan Jokifá	Technology University of Delft, Netherlands
Charlie Joyez	GREDEG, Université Côte d'Azur, France
Bogumil Kamiński	SGH Warsaw School of Economics, Poland
Marton Karsai	Central European University, Austria
Eytan Katzav	Hebrew University of Jerusalem, Israel
Mehmet Kaya	Firat University, Turkey
Domokos Kelen	SZTAKI, Hungary
Mohammad Khansari	Sharif University of Technology, Iran
Jinseok Kim	University of Michigan, USA
Pan-Jun Kim	Hong Kong Baptist University, Hong Kong
Maksim Kitsak	TU Delft, Netherlands
Mikko Kivelä	Aalto University, Finland
Brennan Klein	Northeastern University, UK
Konstantin Klemm	IFISC (CSIC-UIB), Spain
Xiangjie Kong	Zhejiang University of Technology, China
Onerva Korhonen	University of Eastern Finland, Finland
Miklós Krész	InnoRenew CoE, Slovenia
Prosenjit Kundu	DA-IICT, Gandhinagar, Gujarat, India
Haewoon Kwak	Indiana University Bloomington, USA
Richard La	University of Maryland, USA
Josè Lages	Université de Franche-Comté, France
Renaud Lambiotte	University of Oxford, UK
Aniello Lampo	UC3M, Spain
Jennifer Larson	Vanderbilt University, USA
Paul J. Laurienti	Wake Forest, USA
Anna T. Lawniczak	University of Guelph, Canada
Deok-Sun Lee	KIAS, South Korea
Harlin Lee	Univ. of North Carolina at Chapel Hill, USA
Juergen Lerner	University of Konstanz, Germany

Lasse Leskelä	Aalto University, Finland
Petri Leskinen	Aalto University/SeCo, Finland
Inmaculada Leyva	Universidad Rey Juan Carlos, Spain
Cong Li	Fudan University, China
Longjie Li	Lanzhou University, China
Ruiqi Li	Beijing Univ. of Chemical Technology, China
Xiangtao Li	Jilin University, China
Hao Liao	Shenzhen University, China
Fabrizio Lillo	Università di Bologna, Italy
Giacomo Livan	University of Pavia, Italy
Giosue' Lo Bosco	Università di Palermo, Italy
Hao Long	Jiangxi Normal University, China
Juan Carlos Losada	Universidad Politécnica de Madrid, Spain
Laura Lotero	Universidad Nacional de Colombia, Colombia
Yang Lou	National Yang Ming Chiao Tung Univ., Taiwan
Meilian Lu	Beijing Univ. of Posts and Telecom., China
Maxime Lucas	CENTAI, Italy
Lorenzo Lucchini	Bocconi University, Italy
Hanbaek Lyu	UW-Madison, USA
Vince Lyzinski	University of Maryland, College Park, USA
Morten Mørup	Technical University of Denmark, Denmark
Leonardo Maccari	Ca'Foscari University of Venice, Italy
Matteo Magnani	Uppsala University, Sweden
Maria Malek	CY Cergy Paris University, France
Giuseppe Mangioni	University of Catania, Italy
Andrea Mannocci	CNR-ISTI, Italy
Rosario N. Mantegna	University of Palermo, Italy
Manuel Sebastian Mariani	University of Zurich, Switzerland
Radek Marik	CTU in Prague, Czech Republic
Daniele Marinazzo	Ghent University, Belgium
Andrea Marino	University of Florence, Italy
Malvina Marku	INSERM, CRCT, France
Antonio G. Marques	King Juan Carlos University, Spain
Christoph Martin	Hamburg University of Applied Sciences, Germany
Samuel Martin-Gutierrez	Complexity Science Hub Vienna, Austria
Cristina Masoller	Universitat Politecnica de Catalunya, Spain
Rossana Mastrandrea	IMT School for Advanced Studies, Italy
John D. Matta	Southern Illinois Univ. Edwardsville, USA
Carolina Mattsson	CENTAI Institute, Italy
Fintan McGee	Luxembourg IST, Luxembourg
Matus Medo	University of Bern, Switzerland

Ronaldo Menezes	University of Exeter, UK
Humphrey Mensah	Epsilon Data Management, LLC, USA
Anke Meyer-Baese	Florida State University, USA
Salvatore Micciche	UNIPA DiFC, Italy
Letizia Milli	University of Pisa, Italy
Marija Mitrovic	Institute of Physics Belgrade, Serbia
Andrzej Mizera	University of Warsaw, Poland
Chiara Mocenni	University of Siena, Italy
Roland Molontay	Budapest UTE, Hungary
Sifat Afroj Moon	University of Virginia, USA
Alfredo Morales	MIT, USA
Andres Moreira	UTFSM, Chile
Greg Morrison	University of Houston, USA
Igor Mozetic	Jozef Stefan Institute, Slovenia
Sarah Muldoon	State University of New York, Buffalo, USA
Tsuyoshi Murata	Tokyo Institute of Technology, Japan
Jose Nacher	Toho University, Japan
Nishit Narang	NIT Delhi, India
Filipi Nascimento Silva	Indiana University, USA
Muaz A. Niazi	National Univ. of Science & Technology, Pakistan
Peter Niemeyer	Leuphana University Lueneburg, Germany
Jordi Nin	ESADE, Universitat Ramon Llull, Spain
Rogier Noldus	Ericsson, Netherlands
Masaki Ogura	Osaka University, Japan
Andrea Omicini	Università di Bologna, Italy
Gergely Palla	Eötvös University, Hungary
Fragkiskos Papadopoulos	Cyprus University of Technology, Cyprus
Symeon Papadopoulos	Centre for Research & Technology, Greece
Alice Patania	University of Vermont, USA
Leto Peel	Maastricht University, Netherlands
Hernane B. B. Pereira	Senai Cimatec, Brazil
Josep Perelló	Universitat de Barcelona, Spain
Anthony Perez	Université d'Orléans, France
Juergen Pfeffer	Technical University of Munich, Germany
Carlo Piccardi	Politecnico di Milano, Italy
Pietro Hiram Guzzi	Univ. Magna Gracia of Catanzaro, Italy
Yoann Pigné	Université Le Havre Normandie, France
Bruno Pinaud	University of Bordeaux, France
Flavio L. Pinheiro	Universidade Nova de Lisboa, Portugal
Manuel Pita	Universidade Lusófona, Portugal
Clara Pizzuti	CNR-ICAR, Italy

Jan Platos	VSB - Technical University of Ostrava, Czech Republic
Pawel Pralat	Toronto Metropolitan University, Canada
Rafael Prieto-Curiel	Complexity Science Hub, Austria
Daniele Proverbio	University of Trento, Italy
Giulia Pullano	Georgetown University, USA
Rami Puzis	Ben-Gurion University of the Negev, Israel
Christian Quadri	Università degli Studi di Milano, Italy
Hamid R. Rabiee	Sharif University of Technology, Iran
Filippo Radicchi	Indiana University, USA
Giancarlo Ragozini	University of Naples Federico II, Italy
Juste Raimbault	IGN-ENSG, France
Sarah Rajtmajer	Penn State, USA
Gesine D. Reinert	University of Oxford, UK
Élisabeth Remy	Institut de Mathématiques de Marseille, France
Xiao-Long Ren	Univ. of Electronic Science and Tech., China
Laura Ricci	University of Pisa, Italy
Albano Rikani	INSERM, France
Luis M. Rocha	Binghamton University, USA
Luis E. C. Rocha	Ghent University, Belgium
Fernando E. Rosas	Imperial College London, UK
Giulio Rossetti	CNR-ISTI, Italy
Camille Roth	CNRS/CMB/EHESS, France
Celine Rozenblat	UNIL, Switzerland
Giancarlo Ruffo	Univ. degli Studi del Piemonte Orientale, Italy
Arnaud Sallaberry	University of Montpellier, France
Hillel Sanhedrai	Northeastern University, USA
Iraj Saniee	Bell Labs, Nokia, USA
Antonio Scala	CNR Institute for Complex Systems, Italy
Michael T. Schaub	RWTH Aachen University, Germany
Irene Sendiña-Nadal	Universidad Rey Juan Carlos, Spain
Mattia Sensi	Politecnico di Torino, Italy
Ke-ke Shang	Nanjing University, China
Julian Sienkiewicz	Warsaw University of Technology, Poland
Per Sebastian Skardal	Trinity College, Ireland
Fiona Skerman	Uppsala University, Sweden
Oskar Skibski	University of Warsaw, Poland
Keith M. Smith	University of Strathclyde, UK
Igor Smolyarenko	Brunel University, UK
Zbigniew Smoreda	Orange Innovation, France
Annalisa Socievole	ICAR-CNR, Italy
Igor M. Sokolov	Humboldt University Berlin, Germany

Albert Solé-Ribalta	Universitat Oberta de Catalunya, Spain
Sara Sottile	University of Trento, Italy
Sucheta Soundarajan	Syracuse University, USA
Jaya Sreevalsan-Nair	IIIT Bangalore, India
Christoph Stadtfeld	ETH Zürich, Switzerland
Clara Stegehuis	University of Twente, Netherlands
Lovro Šubelj	University of Ljubljana, Slovenia
Xiaoqian Sun	Beihang University, China
Michael Szell	IT University of Copenhagen, Denmark
Boleslaw Szymanski	Rensselaer Polytechnic Institute, USA
Andrea Tagarelli	University of Calabria, Italy
Kazuhiro Takemoto	Kyushu Institute of Technology, Japan
Frank W. Takes	Leiden University, Netherlands
Fabien Tarissan	CNRS & ENS Paris-Saclay, France
Laura Temime	Cnam, France
François Théberge	TIMC, France
Guy Theraulaz	Université Paul Sabatier and CNRS, France
I-Hsien Ting	National University of Kaohsiung, Taiwan
Michele Tizzani	ISI Foundation, Italy
Michele Tizzoni	University of Trento, Italy
Olivier Togni	University of Burgundy, France
Leo Torres	Northeastern University, USA
Sho Tsugawa	University of Tsukuba, Japan
Francesco Tudisco	The University of Edinburgh, UK
Melvyn S. Tyloo	Los Alamos National Lab, USA
Stephen M. Uzzo	National Museum of Mathematics, USA
Lucas D. Valdez	IFIMAR-UNMdP, Argentina
Pim Van der Hoorn	Eindhoven University of Technology, Netherlands
Piet Van Mieghem	Delft University of Technology, Netherlands
Fabio Vanni	University of Insubria, Italy
Christian L. Vestergaard	Institut Pasteur, France
Tiphaine Viard	Télécom Paris, France
Julian Vicens	Eurecat, Spain
Blai Vidiella	CSIC, Spain
Pablo Villegas	Enrico Fermi Research Center (CREF), Italy
Maria Prosperina Vitale	University of Salerno, Italy
Pierpaolo Vivo	King's College London, UK
Johannes Wachs	Corvinus University of Budapest, Hungary
Huijuan Wang	Delft University of Technology, Netherlands
Lei Wang	Beihang University, China
Guanghui Wen	Southeast University, Nanjing, China
Mateusz Wilinski	Los Alamos National Laboratory, USA

Dirk Witthaut	Forschungszentrum Jülich, Germany
Bin Wu	Beijing Univ. of Posts and Telecom., China
Mincheng Wu	Zhejiang University of Technology, China
Tao Wu	Chongqing Univ. of Posts and Telecom., China
Haoxiang Xia	Dalian University of Technology, China
Gaoxi Xiao	Nanyang Technological University, Singapore
Nenggang Xie	Anhui University of Technology, China
Takahiro Yabe	MIT, USA
Kaicheng Yang	Northeastern University, USA
Yian Yin	Cornell University, USA
Jean-Gabriel Young	University of Vermont, USA
Irfan Yousuf	Univ. of Engineering and Technology, Pakistan
Yongguang Yu	Beijing Jiaotong University, China
Paolo Zeppini	University Cote d'Azur, France
Shi Zhou	University College London (UCL), UK
Wei-Xing Zhou	East China Univ. of Science and Techno., China
Eugenio Zimeo	University of Sannio, Italy
Lorenzo Zino	Politecnico di Torino, Italy
Michal R. Zochowski	University of Michigan, USA
Claudia Zucca	Tilburg University, Netherlands

Contents

Community Structure

Identifying Well-Connected Communities in Real-World and Synthetic Networks	3
<i>Minhyuk Park, Yasamin Tabatabaee, Vikram Ramavarapu, Baqiao Liu, Vidya Kamath Pailodi, Rajiv Ramachandran, Dmitriy Korobskiy, Fabio Ayres, George Chacko, and Tandy Warnow</i>	
Bayesian Hierarchical Network Autocorrelation Models for Modeling the Diffusion of Hospital-Level Quality of Care	15
<i>Guanqing Chen and A. James O'Malley</i>	
Topological Community Detection: A Sheaf-Theoretic Approach	29
<i>Arne Wolf and Anthea Monod</i>	
Does Isolating High-Modularity Communities Prevent Cascading Failure?	43
<i>Stephen Eubank</i>	
Two to Five Truths in Non-negative Matrix Factorization	55
<i>John M. Conroy, Neil Molino, Brian Baughman, Rod Gomez, Ryan Kaliszewski, and Nicholas A. Lines</i>	
Adopting Different Strategies for Improving Local Community Detection: A Comparative Study	68
<i>Konstantinos Christopoulos and Konstantinos Tsichlas</i>	
Pyramid as a Core Structure in Social Networks	82
<i>Wenruo Lyu and Liang Zhao</i>	
Dual Communities Characterize Structural Patterns and Robustness in Leaf Venation Networks	95
<i>Philipp C. Böttcher, Franz Kaiser, Henrik Ronellenfitsch, Vito Latora, and Dirk Witthaut</i>	
Tailoring Benchmark Graphs to Real-World Networks for Improved Prediction of Community Detection Performance	108
<i>Catherine Schwartz, Cetin Savkli, Amanda Galante, and Wojciech Czaja</i>	

Network Based Methodology for Characterizing Interdisciplinary Expertise in Emerging Research	121
<i>Aditi Mallavarapu, Erin Walker, Cassandra Kelley, Shari Gardner, Jeremy Roschelle, and Stephen Uzzo</i>	
Classification Supported by Community-Aware Node Features	133
<i>Bogumił Kamiński, Paweł Pralat, François Théberge, and Sebastian Zajac</i>	
Signature-Based Community Detection for Time Series	146
<i>Marco Gregnanin, Johannes De Smedt, Giorgio Gnecco, and Maurizio Parton</i>	
Hierarchical Overlapping Community Detection for Weighted Networks	159
<i>Petr Prokop, Pavla Dráždilová, and Jan Platoš</i>	
Detecting Community Structures in Patients with Peripheral Nervous System Disorders	172
<i>Morteza Hosseinioun, Ali Mohammad Afshin Hemmatyar, Saeid Ahmadifar, Hojjat Samiee, and S. AmirAli Gh. Ghahramani</i>	
Community Detection in Feature-Rich Networks Using Gradient Descent Approach	185
<i>Soroosh Shalileh and Boris Mirkin</i>	
Detecting Strong Cliques in Co-authorship Networks	197
<i>Lukas Papik, Eliska Ochodkova, and Milos Kudelka</i>	
Mosaic Benchmark Networks: Modular Link Streams for Testing Dynamic Community Detection Algorithms	209
<i>Yasaman Asgari, Remy Cazabet, and Pierre Borgnat</i>	
Entropic Detection of Chromatic Community Structures	223
<i>Franck Delaplace</i>	
On the Hierarchical Component Structure of the World Air Transport Network	235
<i>Issa Moussa Diop, Cherif Diallo, Chantal Cherifi, and Hocine Cherifi</i>	
Weighted and Unweighted Air Transportation Component Structure: Consistency and Differences	248
<i>Issa Moussa Diop, Cherif Diallo, Chantal Cherifi, and Hocine Cherifi</i>	
Effects of Null Model Choice on Modularity Maximization	261
<i>Christopher Brissette, Ujwal Pandey, and George M. Slota</i>	

On Centrality and Core in Weighted and Unweighted Air Transport
Component Structures 273
Issa Moussa Diop, Cherif Diallo, Chantal Cherifi, and Hocine Cherifi

Diffusion and Epidemics

New Seeding Strategies for the Influence Maximization Problem 289
*Seok-Hee Hong, Juan Pablo Bonilla Ataides, Rowena Kok,
Amyra Meidiana, and Kunsoo Park*

Effects of Homophily in Epidemic Processes 300
Richard J. La

Human Papillomavirus Co-circulation on a Partially Vaccinated
Partnership Network 312
*Mélanie Bonneault, Maxime Flauder,
Elisabeth Delarocque-Astagneau, Anne C. M. Thiébaud,
and Lulla Opatowski*

Towards the Building of a Surveillance Network for PPR-Like Diseases
in Nigeria: Identifying Potential Sentinel Node in a Partially-Known
Network 325
*Asma Mesdour, Sandra Ijioma, Muhammad-Bashir Bolajoko,
Elena Arsevska, Mamadou Ciss, Mathieu Andraud, Andrea Apolloni,
and Eric Cardinale*

Travel Demand Models for Micro-Level Contact Network Modeling 338
Diaoulé Diallo, Jurij Schönfeld, and Tobias Hecking

Evaluating Attitudes on Health-Seeking Behavior Among a Network
of People Who Inject Drugs 350
*Ayako Shimada, Ashley L. Buchanan, Natallia V. Katenka,
Benjamin Skov, Gabrielle Lemire, Stephen Kogut,
and Samuel R. Friedman*

On the Relation Between Replicator Evolutionary Dynamics and Diffusive
Models on General Networks 362
Rio Aurachman and Giuliano Punzo

Dynamics on/of Networks

SMART CONTRACTS Based Peer to Peer Communication in Blockchain: A Decentralized Approach	373
<i>Saty Bhushan Verma, Abhay Kumar Yadav, Bineet Kumar Gupta, Sanjay Gupta, and Rishi Srivastava</i>	
A Quadratic Static Game Model for Assessing the Impact of Climate Change	383
<i>Bouchra Mroué, Anthony Couthures, Samson Lasaulce, and Irinel Constantin Morărescu</i>	
Linear Stochastic Processes on Networks and Low Rank Graph Limits	395
<i>Alex Duniyakh and Peter E. Caines</i>	
Uniform Generation of Temporal Graphs with Given Degrees	408
<i>Daniel Allendorf</i>	
A Multi-order Adaptive Network Model for Pathways of DNA Methylation and Its Effects in Individuals Developing Post-traumatic Stress Disorder	421
<i>Iva Gunjača, Natalie Samhan, and Jan Treur</i>	
DynamicScore: A Novel Metric for Quantifying Graph Dynamics	435
<i>Bridonneau Vincent, Guinand Frédéric, and Pigné Yoann</i>	
A Novel Method for Vertex Clustering in Dynamic Networks	445
<i>Devavrat Vivek Dabke and Olga Dorabiala</i>	
A Particle Method for Continuous Hegselmann-Krause Opinion Dynamics	457
<i>Christoph Börgers, Natasa Dragovic, Anna Haensch, and Arkadz Kirshtein</i>	
Optimal Reconstruction of Graph Evolution Dynamics for Duplication-Based Models	470
<i>Emre Sefer and Samuel Gilmour</i>	
Farthest-First Traversal for Identifying Multiple Influential Spreaders	484
<i>Madhvi Ramrakhiyani, Mukesh Tiwari, and V. Sunitha</i>	
Wishful Thinking About Consciousness	492
<i>Peter Grindrod</i>	
Author Index	503

Community Structure



Identifying Well-Connected Communities in Real-World and Synthetic Networks

Minhyuk Park¹, Yasamin Tabatabaee¹, Vikram Ramavarapu¹, Baqiao Liu¹,
Vidya Kamath Pailodi¹, Rajiv Ramachandran¹, Dmitriy Korobskiy²,
Fabio Ayres³, George Chacko¹(✉), and Tandy Warnow¹(✉)

¹ University of Illinois Urbana-Champaign, Urbana, IL, USA

{chackoge,warnow}@illinois.edu

² NTT DATA, McLean, VA, USA

³ Insper Institute, São Paulo, Brazil

Abstract. Integral to the problem of detecting communities through graph clustering is the expectation that they are “well-connected”. Surprisingly, we find that the output of multiple clustering approaches—the Leiden algorithm with either the Constant Potts Model or modularity as quality function, Iterative K-Core Clustering, Infomap, and Markov Clustering—include communities that fail even a mild requirement for well-connectedness. As a remediation strategy, we have developed the “Connectivity Modifier” (CM), which iteratively removes small edge cuts and re-clusters until all communities detected are well-connected. Results from real-world networks with up to 75,025,194 nodes illustrate how CM enables additional insights into community structure within networks, while results on synthetic networks show that the CM algorithm improves accuracy in recovering true communities. Our study also raises questions about the “clusterability” of networks and mathematical models of community structure.

Keywords: community detection · connectivity · citation networks

1 Introduction

Community detection is of broad interest and is typically posed as a graph partitioning problem, where the input is a graph and the objective is a partitioning of its vertices into disjoint subsets, so that each subset represents a community [12, 23, 24]. The terms community and cluster overlap heavily, so we use them interchangeably herein. While community detection has many applications [8, 18], our interest is largely related to identifying research communities from the global scientific literature. Accordingly we are especially focused on methods that can scale to large citation networks [36, 37].

A general expectation is that the vertices within a community are better connected to each other than to vertices outside the community [7, 16], implying

M. Park, Y. Tabatabaee, and V. Ramavarapu—Contributed equally.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

H. Cherifi et al. (Eds.): COMPLEX NETWORKS 2023, SCI 1142, pp. 3–14, 2024.

https://doi.org/10.1007/978-3-031-53499-7_1

greater edge density within a community. However, a cluster may be dense while still having a small min cut (the smallest edge set whose removal would disconnect the cluster) [3]. In other words, some dense clusters can be disconnected by small min cuts. Thus, *edge density* and *well-connectedness* are expected but *separable* properties of communities.

The potential for modularity optimization to produce poorly connected clusters, for example, two large cliques connected by a single edge, is well established [11, 36]. Less well studied, however, is the question of whether other clustering methods also produce poorly connected clusters. One of the commonly used clustering methods is the Leiden algorithm [36] optimizing the Constant Potts Model (CPM) [35]. Clusters produced by CPM-optimization with resolution parameter r have the desirable property that if an edge cut splits the cluster into components A and B , then the edge cut size will be at least $r \times |A| \times |B|$ (Supplementary Materials of reference [36]). This guarantee is strong when the edge cut splits a cluster into two components of approximately equal size, but is weaker when it produces an unbalanced split and weakest when the cut separates a single node from the remaining nodes in the cluster. Importantly, the guarantee depends on r , and small values of r produce weak bounds. It is also important to note that this guarantee applies to CPM-optimal clusterings but not to heuristics.

In using the Leiden software optimizing CPM, we observe that it produces clusters with small min cuts on seven different real-world networks of varied origin ranging in size from approximately 34,000 to 75 million nodes. We also observe that the number of clusters with small min cuts increases as the resolution parameter is decreased. Intrigued by this observation, we perform a broader study to evaluate the extent to which clusters produced by algorithms of interest meet even a mild standard for a well-connected cluster.

We formalize the notion of “poorly-connected” clusters by considering functions $f(n)$ with the interpretation that if a cluster of size n has an edge cut of size at most $f(n)$ then the cluster will be considered poorly connected. We want $f(n)$ to grow very slowly so that it serves as a mild bound. We also want $f(n) \geq 1$ for all n that are large enough for the cluster to be considered a potential community. We choose $f(n) = \log_{10} n$ from three examples of slow growing functions [25], since it imposes the mildest constraint on large clusters and grows more slowly than the bound for optimal CPM clusterings [36].

We examine min cut profiles from four additional clustering methods on the seven networks: Leiden optimizing modularity [24]; the k -core based Iterative k -core Clustering (IKC) [37]; and two flow-based methods, Infomap [31] and Markov Clustering (MCL) [9]. All the methods we tested produce poorly connected clusters on these networks, some produce tree clusters, and some even produce disconnected clusters. These observations reveal a gap between the expectation of well-connected clusters and what is actually being produced by these community finding methods.

To address this gap, we have developed the Connectivity Modifier (CM) [29] that takes a clustering as input and recursively removes small edge cuts and reclusters until all clusters are well-connected. Using CM on seven real-world

networks, we demonstrate the insights that CM can provide into community structure in networks. These findings also raise questions about the “clusterability” [22] of networks and whether only portions of a network exhibit community structure. Additional analyses on synthetic networks provide evidence that CM improves community detection accuracy under many conditions [25].

2 Results

2.1 Initial Observations

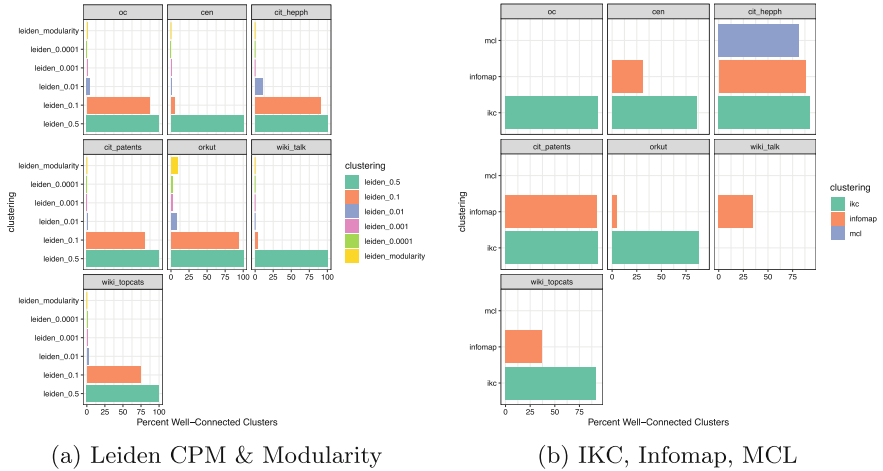


Fig. 1. *Percentage of well-connected clusters in seven real-world networks.* The networks studied range in size from 34,546 nodes to 75,025,194 nodes. Only Leiden and IKC ran to completion on all seven networks. Only Leiden-CPM with the largest tested resolution parameter (0.5) and IKC had 80% or more of their clusters considered well-connected. Five clustering methods were explored: (a) Leiden optimizing CPM at different resolution values and Leiden optimizing modularity, and (b) IKC, Infomap, and MCL. IKC did not return any clusters from the wiki_talk network. Infomap completed on all but Open Citations. MCL completed only on cit_hepph.

In an exploratory experiment, we clustered seven networks (Table 1, Materials and Methods), ranging in size from 34,546 nodes to 75,025,194 nodes, with Leiden, IKC, Infomap, and MCL, and computed the percentage of clusters whose min cuts were greater than $f(n)$. Under the conditions used, Leiden and IKC ran to completion on all seven networks, Infomap failed on the largest network, and MCL returned output only from the smallest network (cit_hepph) we analyzed.

This experiment (Fig. 1) revealed that all clustering methods generate clusters that are not well-connected, with the extent depending on the clustering

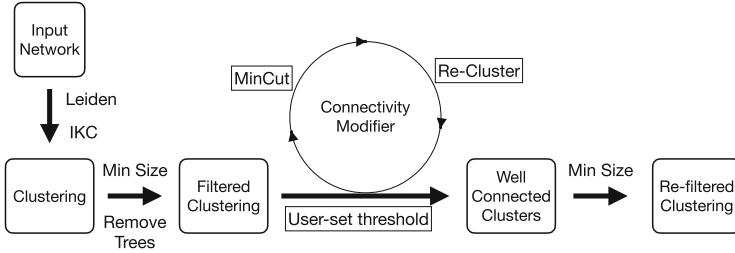


Fig. 2. *Connectivity Modifier Pipeline Schematic.* The four-stage pipeline depends on user-specified algorithmic parameters: B (default 11), the minimum allowed size of a cluster, and $f(n)$ (default $\log_{10}(n)$), a bound on the minimum edge cut size for a cluster with n nodes, and clustering method. *Stage 1:* a clustering is computed. *Stage 2:* clusters are pre-processed by removing trees and those clusters of size less than B . *Stage 3:* the CM is applied to each cluster, removing edge cuts of sizes at most $f(n)$, reclustering, and recursing on clusters. *Stage 4:* clusters are post-processed by removing those of size less than B .

method and network. Most significantly, only IKC and Leiden-CPM at a large resolution value returned a high fraction of well-connected clusters.

For Leiden clustering optimizing CPM, the frequency of well-connected clusters decreases with resolution value, and results from modularity are similar to the lowest resolution value for CPM that was tested. In comparison, nearly all IKC clusters were well-connected, with percentages that varied between 85.9% and 94% of the total number of clusters but with lower node coverage [25]. The percentage of well-connected clusters produced by Infomap varied from 5% (orkut) to 92.4% (cit_patents). For the single network that MCL completed on, 81.3% of the clusters were well-connected. Interestingly, both Infomap and MCL generated clusters that were disconnected, a limitation that had been previously noted for Louvain optimizing modularity [36].

2.2 Connectivity Modifier

To remediate poorly connected clusters, we developed the Connectivity Modifier (CM) [29], which takes a clustering as input and returns well-connected clusters. CM presently provides support for Leiden optimizing either CPM or modularity and IKC, the methods that scaled to the largest network we studied. CM is implemented in a pipeline (Fig. 2), which allows the user to specify two parameters: $f(n)$ (the bound on the size of a min cut) and B , the minimum allowed size of a cluster. In our study we explored $f(n) = \log_{10}(n)$ and $B = 11$, but the user can provide different settings. A pre-processing (filtering) step discards clusters that are trees or of size less than B , noting that any tree with ten or more nodes is not well-connected according to our definition of $f(n)$. CM then checks each cluster to see if it contains an edge cut of size at most $f(n)$, and if so CM removes the edge cut, following which the resultant subnetworks are reclustered. This process repeats until the current iteration produces no change.

A post-processing step removes any small clusters of size less than B that may have resulted from repeated cutting.

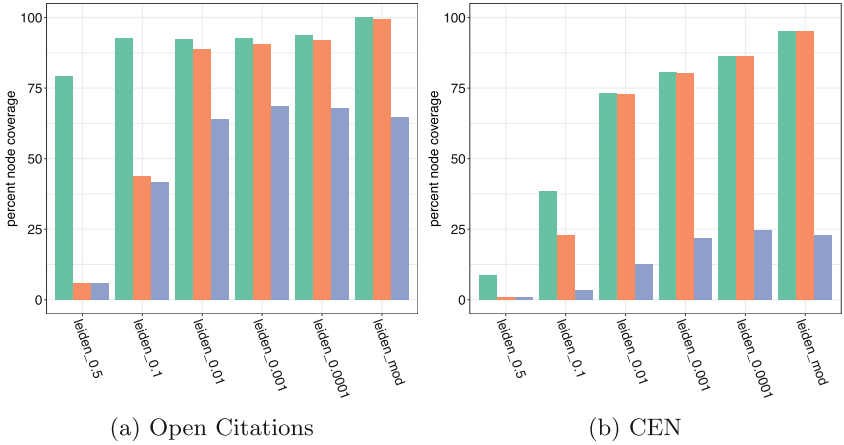


Fig. 3. Reduction in node coverage after CM treatment of Leiden clusters. The Open Citations (left panel) and CEN (right panel) networks were clustered using the Leiden algorithm under CPM at five different resolution values or modularity. Node coverage (defined as the percentage of nodes in clusters of size at least 2) was computed for (i) Leiden clusters (green), (ii) Leiden clusters with trees and clusters of size 10 or less filtered out (orange), and (iii) after CM treatment of filtered clusters (blue).

To further understand the nature of the modifications effected by CM, we also classified the Leiden clusters based on the impact of CM-processing: *extant*, *reduced*, *split*, and *degraded*, where “extant” indicates that the cluster was not modified by CM, “reduced” indicates that the cluster is reduced in size, “split” indicates that the cluster was divided into at least two smaller clusters, and “degraded” indicates that the cluster was reduced to singletons or a cluster of size 10 or less [25]. All methods produced split clusters, suggesting “resolution limit” behavior that has already been documented for modularity [11]. Our study shows this also occurs at some non-negligible frequency for CPM-optimization using small resolution values, as well as for the other clustering methods.

2.3 Effect of CM on Clustered Real World Networks

We studied the effect of CM on clusterings generated by Leiden-modularity, Leiden-CPM, and IKC, the only methods that scaled to the largest network we studied and did not produce disconnected clusters. We present results here from the Open Citations and CEN networks, the two largest networks out of seven studied. Results on the remaining five networks show similar trends [25]. In assessing the impact of CM on node coverage, we do not consider very small clusters ($n \leq 10$) of practical interest, therefore, unless otherwise described, node

coverage is reported in this submission as the percentage of nodes in clusters of at least size 11.

With Leiden, maximum node coverage was 24.6% for the CEN and 68.7% for Open Citations for six conditions tested. Post-CM node coverage was smallest when using CPM-optimization with resolution value $r = 0.5$ and largest when optimizing CPM with one of the two smallest resolution values, $r = 0.001$ for CEN and $r = 0.0001$ for Open Citations. In comparison to Leiden, the IKC-clustering results in relatively low node coverage, 23.6% and 3.8% in the case of the Open Citations and CEN networks respectively. CM treatment of these clusterings has a small effect on node coverage. Results for the other five networks show similar trends and are available at [25].

The impact of CM depends on both the clustering method and the input network. A more detailed examination shows the following trends. For CPM-optimization, the impact of pre-processing is large for the two larger resolution values, but decreases with resolution value. In contrast, the step of the CM pipeline that iteratively finds and removes small edge cuts and reclusters has a minimal impact at larger resolution values and an increasing impact as the resolution value decreases. Modularity returned results most similar to CPM-optimization with the smallest tested resolution value.

These trends suggest the possibility that only a fraction of the nodes in a network belong in clusters that are sufficiently well-connected and sufficiently large. In other words, and through the lens of our mild standard for connectivity and size, real-world networks may not be fully covered by “valid” communities, in contrast to what is often assumed in mathematical models of community structure in networks.

2.4 Synthetic LFR Networks

To evaluate the impact of CM-processing on accuracy, we examined synthetic networks produced using LFR software [19]. For this experiment, we computed statistics for the Leiden clusterings of the seven real-world networks we previously explored, and used them as input to the LFR software [19] (see Materials and Methods). We produced a collection of 34 LFR networks with ground truth communities and clustered each of these 34 LFR networks using Leiden with the same clustering parameters used to provide empirical statistics to LFR.

We examined the impact of CM-processing on clustering accuracy; results for Normalized Mutual Information (NMI) and Adjusted Mutual Information (AMI) are shown in Fig. 4. Adjusted Rand Index (ARI) data are available in supplementary data [25]. CM-processing improves NMI accuracy for modularity and also for CPM-optimization when used with small resolution values. CM-processing tends to be otherwise neutral for NMI, and was only detrimental in two cases. The impact on AMI accuracy is more variable. For example, CM-processing reduced AMI accuracy for all conditions with *wiki_talk* except for CPM-optimization with $r = 0.1$, where accuracy was very low and the impact was neutral. CM-processing also reduced AMI accuracy for CPM-optimization on some network:clustering pairs for large resolution values. However, the cases

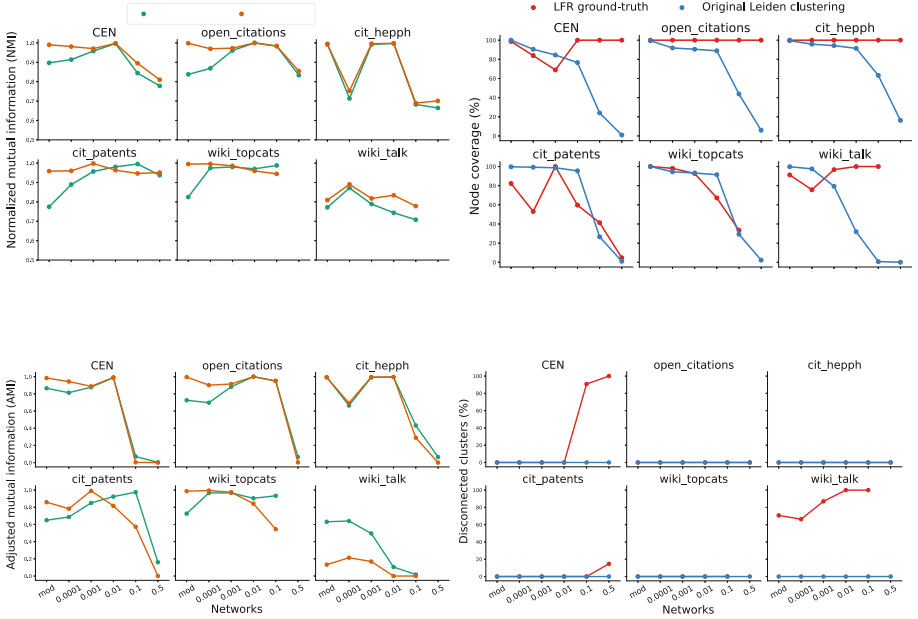


Fig. 4. *Impact of CM-processing on accuracy of synthetic networks.* The left panels show accuracy measured in terms of NMI and AMI with respect to the LFR ground-truth communities. Each condition on the x-axis corresponds to a *different* LFR network, generated based on Leiden-modularity or Leiden-CPM with that specific resolution parameter. The right panels show (top) node coverage when restricted to clusters of size at least 11, and (bottom) percent of clusters that are disconnected. Under most conditions, CM improves the accuracy of the original Leiden clustering, except when the ground-truth communities have many (at least 60%) disconnected clusters, or the node coverage by clusters of size at least 11 is relatively low (at most 70%).

where CM-processing produced a noteworthy reduction in accuracy for NMI or AMI are those where there are many disconnected ground truth clusters or where there are many clusters of size less than 11 in the “ground-truth clusters”. Interestingly, CM can still have a positive impact even when there are many small clusters or some disconnected clusters.

It is easy to see why low node coverage by clusters of size at least 11 could reduce accuracy for CM-processing, since CM with $B=11$ automatically removes all clusters below size 11. Therefore, CM-processing will not be beneficial where there is interest in recovering small communities unless the bound B is replaced by a smaller value. In contrast, the occurrence of disconnected ground-truth clusters in the LFR networks is striking and problematic, since an expectation of a community is that it is connected [36]. Hence, we assert that it is unreasonable to evaluate accuracy with respect to a ground-truth set of communities if the communities are not connected. The fact that LFR networks had ground truth clusters that were not connected also indicates the failure of LFR software to

reproduce features of the input network:clustering pairs, which by construction always have 100% of the clusters connected.

Once the comparison between CM-processed clusters and original clusters is restricted to the LFR networks, which do not have disconnected clusters and which have high node coverage after restriction to clusters of size at least 11), we find that CM ranges from neutral to beneficial in terms of the impact on accuracy. Thus, there is a potential benefit in using CM to modify clusters that are not well-connected.

3 Discussion

In this study we report that multiple clustering methods produce clusters that do not meet a mild standard for well-connectedness. To enforce user-defined standards for connectivity and cluster size, we developed the Connectivity Modifier (CM), an open-source tool [29] that presently provides support for the Leiden and IKC software. We show that the CM enables users to assess hidden community structure within clusters.

How significantly CM-processing changes a given clustering depends on the network itself, as some networks seem to be more impacted by CM-processing. It also depends on the choice of clustering algorithm and the parameter settings used.

An important implication of our study using CM on seven real world communities is the possibility that portions of real world networks may not exhibit robust community structure. A related study using a different approach [22] considers the question of whether a graph can be considered clusterable. Moreover, since many clustering methods aim to maximize node coverage, such approaches are likely to merge weakly connected parts of a network into communities that fail modest standards for connectivity. Thus, there is a trade-off between node coverage and well-connectedness that the user must consider in exploring community structure.

The finding that LFR networks produce different patterns than empirical networks is not surprising. First, the LFR methodology assumes that the degree distribution and cluster size distributions follow a power law, which may not apply well to many real-world networks [2, 5, 28, 33]. Second, the degree distribution and cluster size distributions were imperfectly fitted by LFR software in our study. Another questionable assumption in the LFR methodology is that every node is in a community; our study suggests that this assumption may only be reasonable if the communities can be small and/or poorly connected. Exploring the value of improved generators such as *ABCD_o* [17] is a next step.

Several questions remain unanswered. First, to be able to distinguish between the existence of a community and the ability to detect it with a given clustering method. Second, to assess whether too many or too few nodes are being dropped. While we emphasize leaving the definition, use, and interpretation of well-connected to users, these questions merit attention. We also recognize that our emphasis on well-connected clusters may result in narrow descriptions of

communities. Additionally, informative weaker links [13] may be lost from communities since CM partitions input clusters that are poorly connected into sets of well-connected clusters. On the other hand, these well-connected communities could be useful in defining cores of core-periphery structures [4, 30, 37] that could subsequently be augmented by adding peripheral components. Last, future work should incorporate evaluation criteria relevant to the input network.

Materials and Methods

Table 1. Real world networks used in this study

network	# nodes	# edges	average_node_deg	reference
Open Citations	75,025,194	1,363,303,678	36.34	[27]
CEN	13,989,436	92,051,051	13.16	[15]
cit_hepph	34,546	420,877	24.37	[21]
cit_patents	3,774,768	16,518,947	8.75	[21]
orkut	3,072,441	117,185,083	76.28	[38]
wiki_talk	2,394,385	4,659,565	3.89	[20]
wiki_topcats	1,791,489	25,444,207	28.41	[39]

Data. The publicly available Open Citations dataset was downloaded in Aug 2022. The CEN is a citation network constructed from the literature on exosome research. From the SNAP repository, we downloaded cit_hepph, an High Energy Physics citation network; cit_patents, a citation network of US patents; orkut, a social media network; wiki_talk, a network containing users and discussion from the inception of Wikipedia until January 2008; and wiki_topcats, a web graph of Wikimedia hyperlinks. All networks were processed to remove self-loops, duplicate edges, and parallel edges before clustering.

LFR (Synthetic) Networks. To create simulated networks with ground truth communities, we used the LFR software [10, 19], which takes the following eight parameters as input:

- Network properties: Number of nodes N , average and maximum node degrees (k and k_{max} respectively), and negative exponent for degree sequence (τ_1).
- Community properties: Maximum and minimum community sizes (c_{max} and c_{min}), and negative exponent for the community size distribution (τ_2).
- Mixing parameter μ , that is the ratio between the degree of a node outside its community and its total degree, averaged over all nodes in the network.

To emulate the empirical networks using LFR graphs, we estimated all eight parameters described above for a given pair of network \mathcal{G} and a clustering \mathcal{C} . N, k, k_{max}, c_{min} and c_{max} were computed using *networkX* [14, 34]. To estimate

μ , we performed a single iteration over all edges of the network, and for each edge, if the nodes on the two sides of it were in different communities, that edge contributes to the ratio μ of these two nodes. The total μ of the network:clustering pair is the average μ across all the nodes.

To estimate τ_1 and τ_2 , we fitted a power-law distribution to the node degree sequence and the community size distribution, using the approach from [6] that is implemented in the *powerlaw* Python package [1]. Because the power-law property may not hold for the whole distribution, following [6], we estimated x_{min} , the minimum value for which the power-law property holds as well as the exponent α for the tail of the distribution.

After computing these parameters based on the Leiden clusterings of the empirical networks using both modularity and CPM with a range of resolution parameters, we simulated LFR networks [19]. For networks with more than 10 million nodes, i.e., Open Citations and the CEN, we limited the number of vertices to 3 million, due to scalability limitations of the LFR benchmark graph generator [32], while preserving the edge density reflected by average degree, and the mixing parameter. The numbers of nodes of the other LFR graphs exactly match the number of nodes in the corresponding empirical network. In some cases, due to the inherent limitations of the LFR graph generator, we had to modify the ranges of the community sizes, i.e., increase c_{min} and decrease c_{max} , to generate the network. We calculated NMI, AMI, and ARI using the Python Scikit-Learn package [26].

Acknowledgments and Funding. The authors acknowledge funding from the InSpier-Illinois partnership and an Oracle Research Award to TW. We thank three anonymous reviewers for very constructive critique.

References

1. Alstott, J., Bullmore, E., Plenz, D.: *powerlaw: a Python package for analysis of heavy-tailed distributions*. PloS ONE **9**(1), e85,777 (2014)
2. Artico, I., Smolyarenko, I., Vinciotti, V., Wit, E.C.: How rare are power-law networks really? Proc. Roy. Soc. A **476**(2241), 20190,742 (2020)
3. Bonchi, F., García-Soriano, D., Miyauchi, A., Tsourakakis, C.E.: Finding densest k -connected subgraphs. Discret. Appl. Math. **305**, 34–47 (2021)
4. Breiger, R.: *Explorations in Structural Analysis (RLE Social Theory)*. Routledge, Milton Park (2014)
5. Brzezinski, M.: Power laws in citation distributions: evidence from Scopus. Scientometrics **103**, 213–228 (2015)
6. Clauset, A., Shalizi, C.R., Newman, M.E.: Power-law distributions in empirical data. SIAM Rev. **51**(4), 661–703 (2009)
7. Coscia, M., Giannotti, F., Pedreschi, D.: A classification for community discovery methods in complex networks. Stat. Anal. Data Min. **4**(5), 512–546 (2011)
8. Dey, A.K., Tian, Y., Gel, Y.R.: Community detection in complex networks: from statistical foundations to data science applications. WIREs Comput. Stat. **14**(2) (2021). <https://doi.org/10.1002/wics.1566>

9. Dongen, S.V.: Graph clustering via a discrete uncoupling process. *SIAM J. Matrix Anal. Appl.* **30**(1), 121–141 (2008)
10. Fortunato, S.: Resources (2023). <https://www.santofortunato.net/resources>
11. Fortunato, S., Barthelemy, M.: Resolution limit in community detection. *Proc. Natl. Acad. Sci.* **104**(1), 36–41 (2007)
12. Fortunato, S., Newman, M.E.J.: 20 years of network community detection. *Nat. Phys.* **18**(8), 848–850 (2022)
13. Granovetter, M.S.: The strength of weak ties. *Am. J. Sociol.* **78**(6), 1360–1380 (1973)
14. Hagberg, A., Swart, P., S Chult, D.: Exploring network structure, dynamics, and function using NetworkX. Technical report, Los Alamos National Laboratory (2008)
15. Jakatdar, A., Liu, B., Warnow, T., Chacko, G.: AOC: assembling overlapping communities. *Quant. Sci. Stud.* **3**(4), 1079–1096 (2022)
16. Javed, M.A., Younis, M.S., Latif, S., Qadir, J., Baig, A.: Community detection in networks: a multidisciplinary review. *J. Netw. Comput. Appl.* **108**, 87–111 (2018)
17. Kamiński, B., Pralat, P., Thériberge, F.: Outliers in the ABCD random graph model with community structure (ABCDo). In: Cherifi, H., Mantegna, R.N., Rocha, L.M., Cherifi, C., Micciche, S. (eds.) *COMPLEX NETWORKS 2016 2022*. *SCI*, vol. 1078, pp. 163–174. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-21131-7_13
18. Karataş, A., Şahin, S.: Application areas of community detection: a review. In: *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, pp. 65–70. IEEE (2018)
19. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**(4), 046,110 (2008)
20. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Signed networks in social media. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1361–1370. ACM (2010)
21. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over time: densification laws, shrinking diameters and possible explanations. In: *Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 177–187. ACM (2005). <https://doi.org/10.1145/1081870.1081893>
22. Miasnikof, P., Shestopaloff, A.Y., Raigorodskii, A.: Statistical power, accuracy, reproducibility and robustness of a graph clusterability test. *Int. J. Data Sci. Anal.* **15**(4), 379–390 (2023)
23. Mucha, P.J., Richardson, T., Macon, K., Porter, M.A., Onnela, J.P.: Community structure in time-dependent, multiscale, and multiplex networks. *Science* **328**(5980), 876–878 (2010)
24. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E* **69**(2), 026,113 (2004)
25. Park, M., et al.: Well-connected communities in real-world and synthetic networks (2023). <https://tandy.cs.illinois.edu/cm-for-webpage-combined.pdf>. Includes supplementary materials
26. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
27. Peroni, S., Shotton, D.: OpenCitations, an infrastructure organization for open scholarship. *Quant. Sci. Stud.* **1**(1), 428–444 (2020)
28. Radicchi, F., Fortunato, S., Castellano, C.: Universality of citation distributions: toward an objective measure of scientific impact. *Proc. Natl. Acad. Sci.* **105**(45), 17268–17272 (2008)

29. Ramavarapu, V., Ayres, F., Park, M., Pailodi, V.K., Chacko, G., Warnow, T.: Connectivity modifier (2023). https://github.com/illinois-or-research-analytics/cm_pipeline
30. Rombach, P., Porter, M.A., Fowler, J.H., Mucha, P.J.: Core-periphery structure in networks (revisited). *SIAM Rev.* **59**(3), 619–646 (2017)
31. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.* **105**(4), 1118–1123 (2008)
32. Slota, G.M., Berry, J.W., Hammond, S.D., Olivier, S.L., Phillips, C.A., Rajamanickam, S.: Scalable generation of graphs for benchmarking HPC community-detection algorithms. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–14 (2019)
33. Stringer, M.J., Sales-Pardo, M., Amaral, L.A.N.: Statistical validation of a global model for the distribution of the ultimate number of citations accrued by papers published in a scientific journal. *J. Am. Soc. Inform. Sci. Technol.* **61**(7), 1377–1385 (2010)
34. Tabatabaee, Y.: Emulating real networks using LFR graphs (2023). <https://github.com/ytabatabaee/emulate-real-nets>
35. Traag, V.A., Dooren, P.V., Nesterov, Y.: Narrow scope for resolution-limit-free community detection. *Phys. Rev. E* **84**(1) (2011)
36. Traag, V.A., Waltman, L., Van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**(1), 1–12 (2019)
37. Wedell, E., Park, M., Korobskiy, D., Warnow, T., Chacko, G.: Center-periphery structure in research communities. *Quant. Sci. Stud.* **3**(1), 289–314 (2022)
38. Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. *Knowl. Inf. Syst.* **42**(1), 181–213 (2013)
39. Yin, H., Benson, A.R., Leskovec, J., Gleich, D.F.: Local higher-order graph clustering. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 555–564. ACM (2017)



Bayesian Hierarchical Network Autocorrelation Models for Modeling the Diffusion of Hospital-Level Quality of Care

Guanqing Chen¹ and A. James O'Malley^{2,3}(✉)

¹ Department of Anesthesia, Critical Care and Pain Medicine,
Beth Israel Deaconess Medical Center, Harvard Medical School,
Boston, MA 02215, USA

² The Dartmouth Institute for Health Policy and Clinical Practice,
Geisel School of Medicine at Dartmouth, Lebanon, NH 03756, USA

³ Department of Biomedical Data Science, Geisel School of Medicine at Dartmouth,
Lebanon, NH 03756, USA
James.OMalley@Dartmouth.edu

Abstract. A network autocorrelation model may be embedded within a hierarchical data structure involving a complex network, when the peer effect (also referred to as social influence or contagion) is believed to act between units (e.g., hospitals) above the level at which data is observed. We develop two hierarchical network autocorrelation models to represent peer effects between hospitals when modeling individual outcomes of the patients who attend those hospitals. We use a Bayesian approach for model estimation and assess the performance of the models and sensitivity of results to different prior distributions using a simulation study. We construct a United States New England region patient-sharing hospital network and apply our models and Bayesian estimation procedures to study the diffusion of the adoption of robotic surgery and hospital-peer effects in the quality of patients outcomes using a cohort of United States Medicare beneficiaries in 2016 and 2017.

Keywords: Bayesian inference · Complex network · Diffusion of Robotic surgery · Hierarchical network autocorrelation model · Peer effect

1 Introduction

As one of the most important models in social network analysis, the network autocorrelation model (NAM) has been developed and implemented with the belief that one actor's behavior may be influenced by the behavior of other linked actors in the network [1]. Several types of NAMs have been proposed in the last 40 years. For instance, [2, 3] describe the model: $Y = \rho WY + X\beta + \varepsilon$, $\varepsilon \sim N(0, \sigma^2 I)$, where Y is a vector containing realizations of a dependent variable,

W is a matrix whose elements are social ties between pairs of actors, X is a matrix of covariates and ρ is the level of peer-to-peer influence that underlies diffusion across the network. However, none of these models examined the interdependence among the actors when they are at least one level higher in the data structure than the units on which observations are made. An example in the case of a two-level hierarchical data structure occurs in a study in which interest centers on peer-effects among providers (e.g., physicians or hospitals) and whether patient outcomes are directly impacted by the peers of their provider. Such a study is important as knowing whether a hospital's adoption of a technology impacts the outcomes of a greater population of patients than just their own patients is important for policy-makers to understand in order to make decisions regarding the priority of different incentive programs aiming to improve the quality of patient care and outcomes.

In the similar but distinct area of spatial statistical analysis (spatial data typically has a simpler typology than network data in that distances between points or areas are compliant with the triangle inequality), [4] introduces a family of hierarchical spatial autoregressive models (HSAR) to model hierarchical data structures involving geographic units. The HSAR is given by:

$$Y = \rho WY + X\beta + Z\gamma + \Delta\theta + \varepsilon \text{ with } \theta = \lambda M\theta + u,$$

where X and Z are matrices for the lower and cluster level covariates, θ is a vector of random effects for higher level spatial units, Δ is a design matrix linking θ to Y , and W and M are spatial weight matrices (or neighborhood connection matrices) at the lower and higher levels, respectively. The specification of W and M is naturally based on geographical contiguity (areal spatial data) although the model could be adapted to the case when it is based on geographical distances separating units (point-referenced spatial data). Finally, ρ and λ measure the strength of the dependencies at the observation and the cluster levels, respectively.

In this paper, we first develop the basic hierarchical network autocorrelation model by adapting the HSAR in [4] to social network data assuming the peer effects of actors at a higher level (e.g., hospitals) of the hierarchical structure than the level at which observations are made (e.g., patients). Second, we develop a novel extended hierarchical network autocorrelation model that includes an extra parameter to allow direct across-level influence of actors such as hospitals on individuals (e.g., patients) attributed to other actors. This extended model relaxes the "no direct effect" restriction of the basic HSAR model in which peer hospitals may indirectly impact the patients from the focal hospital through their impact on the focal hospital (i.e., an indirect effect of peer hospitals) but does not allow direct impact of peer hospitals on patients from the focal hospital. The basic HSAR model can be considered as a special case of the extended model in which direct impact is not allowed. We study the first two moments of the observation-level outcomes as a function of these two network autocorrelation parameters to gain insights into the mechanisms that they represent. The adaptation of HSAR to social network data has not been studied in the literature to date while the extension of the model to allow for direct (across-level spillover) effects is an entirely new topic.

Due to the complexity of the data and network structure, we complete Bayesian model specifications so that Bayesian computational methods can be employed to fit the hierarchical network autocorrelation models. A series of simulation studies are conducted to investigate the properties and demonstrate the performance of the resulting estimators under different prior distributions; the sensitivity of posterior inferences to the prior distribution assumed for the peer effect parameter ρ is of particular interest. To alleviate some concerns with commonly-used priors for ρ , we also develop a new prior that imposes uniformity on a natural transformation of ρ .

Motivating this work is an observational study seeking to understand the full impact of the adoption of robotic surgery on the time to discharge from hospital of patients undergoing prostatectomy surgery. Robotic surgery is a robotically-assisted and minimally-invasive procedure in which a surgeon operates robotic arms via a console to perform surgeries. It is believed that the use of robotic surgery is associated with the improvement of patient health outcomes, such as shorter hospital stays, less pain, and lower risk of infection [5]. Robotic surgery is commonly used in prostatectomy for prostate cancer [6] and also assists in the treatment of lung cancer, kidney cancer and colorectal cancer [7–9]. The network upon which we examine peer-effects of is the United States (US) New England region hospital network for patients with prostate, lung, kidney and colorectal cancer and is constructed using the 2016 US fee-for-service Medicare claims data. We study the peer effects among hospitals on prostatectomy time to discharge post-surgery of US Medicare patients in 2017 to allow a lagged peer-effect of network interdependency and to partially protect inferences against reverse causality.

The estimation of the above-described peer-effects of robotic-surgery will advance our understanding of the diffusion of robotic surgery, potentially providing policy makers the insights needed to provide incentives for the adoption of other medical technologies or to decide which ones to prioritise. In general, the extended hierarchical network autocorrelation model will provide insights into whether a hospital’s adoption of technologies have a general beneficial effect on patients in a local area (e.g., by improving general standards of surgical quality including strengthening infection control measures in emergency rooms) such that patients who receive surgeries at other hospitals also benefit. The results will be of potential use to policy-makers hoping to provide incentives to hospitals to adopt technologies that are beneficial to patients.

The remainder of the paper is organized around three main methodological contributions and their illustration in the motivating application. In Sect. 2 we specify notation and develop the basic and extended hierarchical network autocorrelation models assuming the peer effects of actors are at a higher level of the hierarchical structure than the level at which observations are made and allowing direct and indirect across-level peer effects of actors on individuals attributed to actors. The end of Sect. 2 contains the development and in depth exploration of the marginal mean and variance of the expected value of the outcome as a function of the network parameters for both models while in Sect. 3 we specify

prior distributions, including a new transformed uniform prior distribution for ρ , and describe our Bayesian estimation approach. Section 4 consists of a simulation study that includes the sensitivity of posterior inferences to the prior distribution assumed for ρ . The robotic surgery motivating example and the interpretation of the results are presented in Sect. 5, while the paper concludes with a discussion in Sect. 6.

2 Notation and Models

2.1 Hierarchical Network Autocorrelation Model

We adapt the HSAR presented in [4] and introduce the underlying hierarchical model to illustrate the interdependency of actors at a higher-level of the data assuming first that no direct interdependency exists at the observational level. That is, peer-effects only act on individual subjects generating observations through their impact on the cluster-effects of the units (the network actors) in which the individuals are grouped, such as in the following model:

$$\begin{aligned} Y &= Z\theta + B\delta + \varepsilon \\ \delta &= \rho W\delta + X\beta + \tau \end{aligned} \tag{1}$$

where $\varepsilon \sim N(0, \sigma^2 I_N)$, $\tau \sim N(0, \omega^2 I_g)$, Y is a vector of length N containing the values of a response variable for N observations, Z is a $N \times k$ matrix for k observation level covariates whose first column is a vector of $\mathbf{1}_s$ corresponding to the intercept parameter, X is a $g \times l$ matrix for l cluster level covariates, δ is a vector of length g representing the random effect of network actors and B is a $N \times g$ matrix linking the random effect δ back to Y . In addition, ε and τ represent the errors at the observational and cluster levels and W is a $g \times g$ matrix quantifying the relationships between the actors in the associated network. The ij^{th} entry of W , W_{ij} , represents the influence of actor j on actor i . The term “ego” refers to the focal actor being studied while the term “alter” refers to the actors with network edges with the ego, also referred to as “peers”.

The matrix W is constrained to be a non-negative row-normalized matrix, reflecting the absence of negative influences and that relative exposures are the conduit through which social influence transmits. The diagonal of W consists of zeros as self-ties are not permitted in the network. The focal parameter ρ is the peer effect corresponding to the indirect effect of alters (actors to which the focal actor is connected) on the outcomes of the individuals of other actors.

Letting $A = I_g - \rho W$, to ensure A is non-singular and the determinant of A , $|A| \neq 0$, the range of ρ needs to be restricted. Following [10–12], we restrict the parameter space of ρ to $(1/\lambda_{\min}, 1/\lambda_{\max})$, where λ_{\max} and λ_{\min} are the maximum and minimum eigenvalues of the row-normalized W . For a row-normalized W , $1/\lambda_{\max} = 1$ and $1/\lambda_{\min} \leq -1$ [13] with the value of $1/\lambda_{\min}$ becoming more negative with increasing network density. Network density equals $M/(g(g-1))$ for directed networks and $2M/(g(g-1))$ for undirected networks, where (M) is the number of observed ties and $g(g-1)$ is the number of possible ties.

We compute the marginal mean and variance of Y to explore and interpret the peer effect. If A is non-singular, the marginal mean and variance satisfy:

$$\begin{aligned} E(Y) &= Z\theta + BA^{-1}X\beta \\ \text{var}(Y) &= BA^{-1}\omega^2I_gA^{-1T}B^T + \sigma^2I_N \end{aligned}$$

Applying the Neumann series, when the norm of ρW , $|\rho W| < 1$, it follows that:

$$A^{-1} = I_g + (\rho W) + (\rho W)^2 + \dots + (\rho W)^N + \dots = \sum_{h=0}^{\infty} (\rho W)^h$$

The above demonstrates that both the marginal mean and variance of the model depend on ρ and that A^{-1} is a high-order polynomial function of ρ and W .

2.2 Extended Hierarchical Network Autocorrelation Model

A restriction on the model in (1) is that conditional on δ_i there is no direct dependence between the vector of observations Y_j and Y_i for actors j and i , respectively, for any $j \neq i$. In practice, one could imagine situations in which alters may directly influence the individuals associated with the ego in ways other than their impact on the ego. Such situations may arise when a network at the observational level is unmeasured. For example, the patients of one hospital may benefit from improved quality of care at a peer hospital through the patients of the peer-hospital incentivising better health behaviors in the patients of another hospital, or the peer-hospital directly impacting the patients of another hospital by sharing resources. To allow for this possibility, we introduce an extended hierarchical network autocorrelation model with an extra parameter quantifying direct cross-level influence of hospitals on patients of other hospitals:

$$\begin{aligned} Y &= Z\theta + B[\delta + \alpha W_1\delta] + \varepsilon \\ \delta &= \rho W_2\delta + X\beta + \tau \end{aligned} \tag{2}$$

where $\varepsilon \sim N(0, \sigma^2 I_N)$, $\tau \sim N(0, \omega^2 I_g)$, and α is an unrestricted parameter that quantifies the direct network effect of alters on the outcome of individuals from the ego. The matrices W_1 and W_2 could represent different types of relationships between actors; e.g., W_1 could be built on geographical distances between hospitals while W_2 could be built on patient-sharing information between hospitals. With only a single source of network relationship information, in our study we set $W_1 = W_2 = W$. Model 1 is the special case of model 2 in which $\alpha = 0$. Letting $G = B[I_g + \alpha W]$, we compute the marginal mean and variance of Y under (2):

$$\begin{aligned} E(Y) &= Z\theta + GA^{-1}X\beta \\ \text{var}(Y) &= GA^{-1}\omega^2I_gA^{-1T}G^T + \sigma^2I_N \end{aligned}$$

and to help interpret α and ρ as well as distinguish the model in (2) from that in (1), we evaluate these expressions across a range of values of α and ρ .

2.3 Illustration of Marginal Mean and Variance of Extended Model with Simulated Data

We simulated 100 datasets under the extended model assuming a network containing 50 hospitals and 30 individuals per hospital. To determine how the marginal mean and variance of the model change with increasing ρ , we plot the average of the mean of the elements of $E(Y)$ and the average of the diagonal elements of $\text{var}(Y)$ over 100 drawn values on the vertical-axis against ρ on the horizontal-axis (Fig. 1.a. and Fig. 1.b.). Similarly, we evaluate the relationship between α and the marginal mean and variance of the model (Fig. 1.c. and 1.d.). Finally, we investigate the association between the network density d and the marginal mean and variance of the model (Fig. 1.e. and 1.f.).

Figure 1.a. and Fig. 1.b. show that the magnitude of the marginal average mean and variance of Y increases when the value of ρ increases and accelerates exponentially upwards when ρ approaches its upper boundary of 1. When ρ approaches 1, the determinant of A , $|A|$, is close to zero and the entries of A^{-1} are relatively large resulting in extreme exponential behavior. Similar results are found for negative values of ρ . Figure 1.c. reveals a linear decreasing association between the marginal average mean of Y and α while Fig. 1.d. shows that the corresponding marginal variance increases with α . From Fig. 1.e. and Fig. 1.f., we find that the marginal mean and variance display volatile behavior when the network density is smaller than 0.1. When the network density is small, e.g., $d < 0.1$, the simulated network often contains isolated nodes. To overcome the computational issues in matrix row-normalization that occur with isolated nodes, we assume that isolates are equally influenced by all other actors in our specification of W . The volatile behavior of the marginal mean and variance of Y as density approaches 0 is due to the rapid escalation in the frequency of isolates and their resulting impartiality with respect to peer-influence from all other actors in the network.

3 Bayesian Hierarchical Network Autocorrelation Model and Estimation

Under model 1, the likelihood function is given by:

$$f(Y | \theta, \delta, \sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\left(-\frac{(Y - Z\theta - B\delta)^T(Y - Z\theta - B\delta)}{2\sigma^2}\right),$$

and the conditional prior distribution of δ as:

$$p(\delta | \beta, \omega^2, \rho) = |A| (2\pi\omega^2)^{-\frac{q}{2}} \exp\left(-\frac{(A\delta - X\beta)^T(A\delta - X\beta)}{2\omega^2}\right)$$

Due to the bounds on the range of values that ρ may take, the prior distribution of ρ may have a nontrivial impact on the posterior distribution and ensuing posterior inferences. To further investigate prior sensitivity and its influence on the

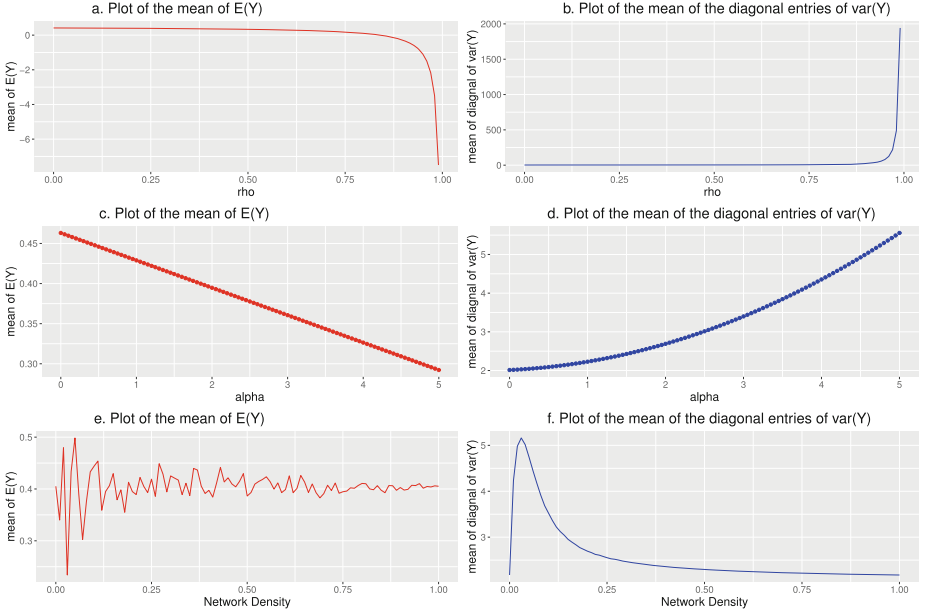


Fig. 1. The marginal mean and variance of the model along with the change of ρ , α and network density

posterior distribution, we specify three priors for ρ and compare the resulting posterior inferences under them. First, we use the flat prior $p(\rho) \propto 1$ over the range $1/\lambda_{\min} < \rho < 1/\lambda_{\max}$ to ensure that the matrix A is non-singular. Second, we retain a uniform prior for ρ but truncate its range to $(-1, 1)$, a symmetric and more restricted parameter space given that $1/\lambda_{\max} = 1$ and $1/\lambda_{\min}$ becomes much smaller than -1 when network density increases. Third, to develop a procedure that emulates the popular Jeffery’s prior family of non-informative seeking prior distributions, we impose prior uniformity on a parameter corresponding to a transformation of ρ whose range is the entire real line. Using the generalized logit function, we assign an improper flat prior on the transformed parameter

$$g(\rho) = \log \left(\frac{1/\lambda_{\max} - \rho}{\rho - 1/\lambda_{\min}} \right)$$

and then derived the implied prior for ρ :

$$p(\rho) \propto \frac{1}{(1/\lambda_{\max} - \rho)(\rho - 1/\lambda_{\min})}$$

which has positive support for $\rho \in (1/\lambda_{\min}, 1/\lambda_{\max})$. In contrast to the flat prior for ρ , the “transformed uniform” prior is “U-shaped” and has more prior mass at its boundary values $\{1/\lambda_{\min}, 1/\lambda_{\max}\}$ than near its center. To complete the prior specification, we assign improper flat priors on σ and ω that are equivalent to

$(\sigma^2, \omega^2) \propto 1/(\sigma\omega)$. (Alternatively, a half Cauchy prior can be used for ω^2 [14]; the results using the half Cauchy prior are essentially indistinguishable from those for the improper flat prior on ω in our analysis.) We specify the flat prior $p(\theta, \beta) \propto 1$ for (θ, β) , although similar results are found from assigning normal priors centered at 0 with large variances (“non-informative normal priors”) for θ and β .

Due to the complexity of the model and the large number of parameters, the joint posterior distribution is non-standard and direct sampling from it is intractable. Therefore, we use a hybrid Gibbs-sampling Metropolis-Hastings algorithm that sequentially draws from the conditional posterior distribution of each parameter given the data and current values of all other parameters [15]. The conditional posterior distributions for each of $\sigma^2, \omega^2, \beta, \theta$ and δ have well-known conjugate forms making sampling from them straightforward. In contrast, the conditional posterior of ρ :

$$p(\rho \mid \beta, \omega^2, \delta, \sigma^2, \theta, Y) \propto |A| \exp\left(-\frac{(A\delta - X\beta)^T(A\delta - X\beta)}{2\omega^2}\right) p(\rho) \quad (3)$$

does not have a form conducive for direct sampling. Therefore, in (3) we approximate $\ln |A|$ using a quadratic polynomial Taylor series approximation at $\rho = 0$ and use a Metropolis Hastings step with an independent candidate generating function for direct sampling [11]. As demonstrated in the derivation in the Supplementary Material in our GitHub site specified at the end of the paper, the resulting candidate generating distribution of ρ when $p(\rho) \propto 1$, $1/\lambda_{\min} < \rho < 1/\lambda_{\max}$, is the truncated normal distribution (TN):

$$\begin{aligned} p(\rho \mid \beta, \omega^2, \delta, \sigma^2, \theta, Y) &\sim TN(\mu^*, V^*) \text{ for } 1/\lambda_{\min} < \rho < 1/\lambda_{\max} \text{ with} \\ \mu^* &= \frac{\delta^T W^T (\delta - X\beta)}{\omega^2 \sum \lambda_i^2 + \delta^T W^T W \delta} \\ V^* &= \frac{\omega^2}{\omega^2 \sum \lambda_i^2 + \delta^T W^T W \delta} \end{aligned} \quad (4)$$

where λ_i for $i = 1, \dots, g$ are the eigenvalues of W . Under $p(\rho) \propto 1$, $-1 < \rho < 1$, the implied candidate generating distribution of ρ is $TN(\mu^*, V^*)$ with support $-1 < \rho < 1$. For the transformed uniform prior of ρ , we use the same candidate generating distribution in Eq. (4) to sample ρ .

Similar to model 1, we use the same prior distributions and MCMC sampling procedure to fully specify and estimate model 2. In addition, we assign a flat prior to the additional parameter α ; that is, $p(\alpha) \propto 1$. The conditional posterior of α is then:

$$p(\alpha \mid \beta, \omega^2, \rho, \sigma^2, \theta, Y, \delta) \sim N\left(\frac{\delta^T W^T B^T K}{\delta^T W^T B^T B W \delta}, \frac{\sigma^2}{\delta^T W^T B^T B W \delta}\right),$$

where $K = Y - Z\theta - B\delta$. As with model 1, the derivation of the conditional posteriors of all other parameters are presented in the Supplemental Materials (see link to our GitHub site at the end of the paper).

4 Simulation Study

We conducted a simulation study involving hypothetical patients receiving care from hospitals in a hospital network to evaluate the performance of model 1 and 2 and their Bayesian estimation procedures under different priors for ρ . The number of hospitals was set to 50 with 30 individuals per hospital. We generated undirected binary-valued network matrices W with density of 0.2, 0.4, 0.6 and 0.8, $\rho = -0.5, -0.2, 0, 0.2, 0.5$ and $\alpha = 2, 5$. For each model, three patient-level and hospital-level covariates were included. For each scenario, we generated 500 simulated datasets and for each drew 20,000 samples from the fitted posterior distribution.

Table 1. Bias, mean squared error (MSE), and 95% coverage rates (Rate) of ρ using uniform priors (Unif 1 for $1/\lambda_{\min} < \rho < 1/\lambda_{\max}$ and Unif 2 for $-1 < \rho < 1$) and transformed uniform prior (T Unif) and α .

Network Density (d)		$\rho = -0.2$			$\rho = 0$			$\rho = 0.2$		
		Unif 1	Unif 2	T Unif	Unif 1	Unif 2	T Unif	Unif 1	Unif 2	T Unif
$d = 0.2$	Bias of ρ	0.024	0.049	0.069	0.010	0.009	0.071	0.002	0.022	0.071
	MSE of ρ	0.059	0.057	0.074	0.053	0.050	0.073	0.049	0.055	0.068
	Rate of ρ	0.958	0.966	0.934	0.962	0.960	0.926	0.952	0.932	0.920
	Bias of α	-0.046	0.035	-0.043	0.005	-0.022	-0.048	0.042	0.018	-0.053
	MSE of α	4.211E-06	2.421E-06	3.761E-06	5.278E-08	9.363E-07	4.663E-06	3.484E-06	6.321E-07	5.711E-06
	Rate of α	0.942	0.956	0.934	0.958	0.956	0.928	0.946	0.946	0.934
$d = 0.4$	Bias of ρ	-0.017	0.134	0.230	√0.105	0.021	0.155	-0.164	-0.123	0.074
	MSE of ρ	0.212	0.145	0.394	0.232	0.125	0.298	0.204	0.135	0.217
	Rate of ρ	0.980	0.984	0.932	0.972	0.984	0.938	0.952	0.980	0.930
	Bias of α	0.103	0.119	0.203	-0.071	0.204	0.236	0.189	0.516	0.267
	MSE of α	2.112E-05	2.837E-05	8.276E-05	1.016E-05	8.332E-05	1.112E-04	7.146E-05	5.329E-04	1.429E-04
	Rate of α	0.972	0.978	0.920	0.974	0.978	0.922	0.958	0.968	0.934

Note: For each value of ρ , the results represent the bias, MSE and Rate of ρ and α . The results are rounded to 3 decimal places.

In general, the simulations reveal that our posterior median estimators of ρ have minimal bias across the three prior distributions with small differences in the corresponding mean squared error (MSE) and the coverage rate of the 95% equal-tailed credible interval. For example, under model 2 the bias of the posterior median estimator of ρ when the network density $d = 0.2$, the indirect peer-effect parameter $\rho = 0.2$, the direct peer-effect parameter $\alpha = 2$, and the prior distribution for ρ is uniform over the full range of possible values of ρ is 0.002 (Table 1). Under the same settings but when $d = 0.4$, bias is -0.164 . To provide a sample of the results we generated in studying the estimators, results for the estimators of ρ and α under model 2 for all combinations of $d = 0.2, 0.4$, $\rho = -0.2, 0, 0.2$, and the three prior distributions when $\alpha = 2$ are also shown in Table 1. Consistent findings are observed for other values of ρ and α under model 1.

We also found that the bias of the posterior median estimators of ρ and α increase with increasing network density, an observation consistent with the findings for classic linear NAMs in [11, 16]. The MSE of ρ and α also increase with increasing network density but this is largely a consequence of the trend for bias as it dominates the MSE. As expected, due to the asymmetric interval of support for ρ , $1/\lambda_{\min} < \rho < 1/\lambda_{\max}$, the uniform prior of ρ leads to the posterior median estimator of ρ exhibiting an asymmetric bias pattern either side of 0. In contrast, bias is much more symmetric around 0 under the uniform $(-1, 1)$ prior for ρ , especially when network density is large. Additionally, for $\rho > 0$ and a large network density, the bias for the transformed uniform prior of ρ is significantly smaller than that under the uniform priors on ρ itself, implying that the mass of the posterior distribution for ρ is pulled more towards its upper boundary under the transformed uniform prior.

Across many settings of the simulation parameters, we have observed similar results implying that as network density increases, it becomes more challenging for the model to identify ρ (see GitHub site for detailed full results). Intuitively, as density increases the information in the data about ρ declines due to the vast number of connections in the network making the variation across the actors in the extent to which they are more or less connected with other actors much lower than when density is low. It is in this high density scenario that slight differences in the non-informative prior specification for ρ nontrivially impact the posterior distribution.

5 The Impact on Patient Quality of Hospitals’ Adoption of Robotic Surgery

To explore whether the extent to which peer-hospital adoption of robotic surgery is associated with the prostatectomy time to discharge post-surgery at a hospital, we construct a US New England region patient-sharing hospital network comprising the six Northeastern US states (Maine, New Hampshire, Vermont, Massachusetts, Connecticut, and Rhode Island) for patients with prostate, lung, kidney and colorectal cancer by adapting the approach in [17–19]. The resulting undirected weighted hospital network matrix is row normalized to form the weight matrix used in models 1 and 2 (the network density is 0.779 with $1/\lambda_{\max} = 1$ and $1/\lambda_{\min} = -1.660$). US Medicare fee-for-service health insurance claims data from 2016 were used to build the patient-sharing hospital network and to evaluate hospital covariates while the corresponding 2017 Medicare data is used to evaluate all patient outcomes and covariates. We include patient’s age, disability, whether receiving a robotic surgery and the Charlson Comorbidity Index [20] as patient-level covariates. For the hospital-level covariates, we include the number of beds, percentage of robotic prostatectomy and number of peer hospitals in the network as covariates. The resulting cohort contains 45 hospitals and 1306 patients.

We use the posterior median as our Bayesian point estimator and compute 95% equal-tailed credible intervals of ρ and α . In addition, we compute the

Deviance Information Criterion (DIC) [21] for model comparison due to its self-determined evaluation of the effective degrees-of-freedom of the model to penalize our Bayesian hierarchical models for model complexity and thus guard against over-fitting when comparing the extended to the basic hierarchical network autocorrelation model.

Table 2. Estimates, credible interval and DIC for model 1 and 2.

Predictors and Key Model Parameters	Estimate (95% Equal-tailed Credible Interval)	
Intercept	0.971(0.922, 1.021)	0.973(0.921, 1.029)
Whether done by robotic surgery	-0.164(-0.214, -0.114)	-0.164(-0.214, -0.114)
Age	0.052(0.029, 0.076)	0.052(0.029, 0.076)
Disability	0.184(0.112, 0.256)	0.184(0.112, 0.257)
Charlson Comorbidity Index	0.145(0.046, 0.244)	0.144(0.045, 0.243)
Beds	-0.015(-0.039, 0.008)	-0.016(-0.045, 0.007)
Percentage of robotic prostatectomy	0.004(-0.018, 0.027)	0.006(-0.016, 0.029)
Number of peer hospitals	0.010(-0.013, 0.032)	0.007(-0.017, 0.031)
ρ (peer effect)	-0.048(-1.164, 0.771)	-0.525(-1.481, 0.804)
α (incremental direct effect of peers on outcome)	NA	1.355(-2.539, 4.168)
σ^2 (residual variance)	0.127(0.117, 0.137)	0.127(0.117, 0.137)
ω^2 (variance of random effects)	1.673E-04 (3.465E-07, 1.772E-03)	2.150E-04 (2.663E-07, 2.173E-03)
DIC (Deviance Information Criterion)	1018.878	1016.843

Note: The results are for the prior $p(\rho) \propto 1$, $1/\lambda_{\min} < \rho < 1/\lambda_{\max}$; similar findings are observed for the other two priors. Numbers are rounded to 3 decimal places.

Table 2 indicates that robotic surgery and number of beds are negatively associated with patients' time to discharge post prostatectomy surgery (i.e., shorter hospital stays occur). In contrast, age, disability, Charlson Comorbidity index, percentage of robotic prostatectomy procedures and the number of peer hospitals are positively associated with time to discharge post-prostatectomy of patients (i.e., longer hospital stays occur). Comparing the two models, we observe a significant change in the magnitude of ρ . With the inclusion of α ($\hat{\alpha} = 1.355(-2.539, 4.168)$), $\hat{\rho}$ changes from -0.048 to -0.525 . The value $\hat{\alpha} = 1.355 > 0$ suggests that peer hospitals' propensity to adopt robotic surgery is directly associated with longer patient hospital stays whereas $\hat{\rho} = -0.525 < 0$ indicates that peer hospitals' propensity to adopt robotic surgery is indirectly associated with shorter hospital stays. The wide credible intervals for both ρ and α overlap 0 and imply that with a very large network density (i.e., 0.779) in our study, the information in the data about ρ and α is limited (much more so than if density were lower). The two models have very close DICs with the DIC of model 2 slightly smaller than the DIC for model 1, suggesting that model 2 fits the data better.

6 Discussion

In this paper, we extended the classic linear NAM to develop two hierarchical network autocorrelation models to study the direct and indirect peer-effects of actors at a higher level of the data than that at which observations are made. The novel contributions include the exploration of peer effects among higher-level actors and the impact of peer-actor behavior on an observation-level outcome within a two-level hierarchical data structure. A special case of our models is the spillover effects model in which network diffusion occurs when an individual's behavior is influenced by their alters' covariates. In addition, we proposed a Bayesian approach for estimation and compared the performance of the resulting estimators under different prior distributions for ρ to gain insights into which prior leads to the most stable posterior inferences and the extent to which the posterior distribution is sensitive to the prior. In general, our model is designed for the analysis of hierarchical models when actor interdependency or peer-effects occurs at a higher-level of the model than that at which observations are made. Although we focused on continuous outcomes, a natural avenue for further research is to generalize the hierarchical and extend hierarchical network autocorrelation models to non-continuous outcomes. Although our simulation study confirmed that our models are estimable, further study of the relationship between network features and the precision of estimation of peer-effects is warranted.

Our model and methodological development were applied to data from an observational study of the diffusion of the adoption of robotic surgery on the quality of patient outcomes. We specifically investigated whether hospital peer effects regarding the adoption of robotic surgery are associated with patient time to discharge post prostatectomy. Although our findings were indecisive, a consequence of insufficient information in the data about the peer-effect parameters due to the densely connected network, in general our models have the potential to be widely applied and to reveal important scientific findings in relation to the impact on the outcomes of patients at a hospital of the adoption of a health technology by its peer hospitals, from which important policy recommendations may be derived.

Supplementary Materials

The data used for the motivating analyses contain patient identifiable information and so cannot be made available. However, template R code for performing the simulations (which can be easily adapted to analyze a real data set) have been uploaded to the paper's GitHub site at: <https://github.com/chen918/HNAM>. The derivation of the conditional posterior distributions and results for the simulation study are also available on the GitHub site.

Acknowledgments. The authors thank Daniel J. Gottlieb and Devang Agravat for helping with data preparation and answering questions about the data. The research in this paper was supported by NIH grant P01 AG019783 and the Agency for Healthcare

Research and Quality's (AHRQ's) Comparative Health System Performance Initiative under Grant # 1U19HS024075.

References

1. O'Malley, A.J., Marsden, P.V.: The analysis of social networks. *Health Serv. Outcomes Res. Methodol.* **8**(4), 222–269 (2008)
2. Doreian, P.: Linear models with spatially distributed data: spatial disturbances or spatial effects? *Sociol. Methods Res.* **9**, 29–60 (1980)
3. Friedkin, N.E.: Social networks in structural equation models. *Soc. Psychol. Q.* **53**(4), 316 (1990)
4. Dong, G., Harris, R.: Spatial autoregressive models for geographically hierarchical data structures. *Geogr. Anal.* **47**(2), 173–191 (2015)
5. Barbash, G.I., Glied, S.A.: New technology and health care costs - the case of robot-assisted surgery. *N. Engl. J. Med.* **363**(8), 701–704 (2010)
6. Lee, D.I.: Robotic prostatectomy: what we have learned and where we are going. *Yonsei Med. J.* **50**(2), 177–181 (2009)
7. Chandra, A., Snider, J.T., Wu, Y., Jena, A., Goldman, D.P.: Robot-assisted surgery for kidney cancer increased access to a procedure that can reduce mortality and renal failure. *Health Affairs* **34**(2), 220–228 (2015)
8. Mirnezami, A.H., Mirnezami, R., Venkatasubramanian, A.K., Chandrakumaran, K., Cecil, T.D., Moran, B.J.: Robotic colorectal surgery: hype or new hope? A systematic review of robotics in colorectal surgery. *Colorectal Dis.* **12**(11), 1084–1093 (2010)
9. Novellis, P., Alloisio, M., Vanni, E., Bottoni, E., Cariboni, U., Veronesi, G.: Robotic lung cancer surgery: review of experience and costs. *J. Visual. Surg.* **3**, 39 (2017)
10. Anselin, L.: *Spatial Econometrics: Methods and Models*. Springer, Dordrecht (1988)
11. Dittrich, D., Leenders, R.Th.A.J., Mulder, J.: Bayesian estimation of the network autocorrelation model. *Soc. Netw.* **48**, 213–236 (2017)
12. LeSage, J.P.: Bayesian estimation of limited dependent variable spatial autoregressive models. *Geogr. Anal.* **32**(1), 19–35 (2000)
13. Stewart, W.J.: *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*. Princeton (2009)
14. Gelman, A.: Prior distribution for variance parameters in hierarchical models. *Bayesian Anal.* **1**(3), 515–534 (2006)
15. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**(6), 721–741 (1984)
16. Mizuchi, M.S., Neuman, E.J.: The effect of density on the level of bias in the network autocorrelation model. *Soc. Netw.* **30**(3), 190–200 (2008)
17. Moen, E.L., Austin, A.M., Bynum, J.P., Skinner, J.S., O'Malley, A.J.: An analysis of patient-sharing physician networks and implantable cardioverter defibrillator therapy. *Health Serv. Outcomes Res. Methodol.* **16**(3), 132–153 (2016)
18. O'Malley, A.J., Moen, E.L., Bynum, J.P.W., Austin, A.M., Skinner, J.S.: Modeling peer effect modification by network strength: the diffusion of implantable cardioverter defibrillators in the US hospital network. *Stat. Med.* **39**(8), 1125–1144 (2020)

19. Bynum, J.P.W., Bernal-Delgado, E., Gottlieb, D., Fisher, E.: Assigning ambulatory patients and their physicians to hospitals: a method for obtaining population-based provider performance measurements. *Health Serv. Res.* **42**(1 Pt 1), 45–62 (2007)
20. Charlson, M., Szatrowski, T.P., Peterson, J., Gold, J.: Validation of a combined comorbidity index. *J. Clin. Epidemiol.* **47**(11), 1245–1251 (1994)
21. Spiegelhalter, D.J., Best, N.G., Carlin, B.P., Van Der Linde, A.: Bayesian measures of model complexity and fit. *J. Roy. Stat. Soc. Ser. B: Stat. Methodol.* **64**(4), 583–639 (2002)



Topological Community Detection: A Sheaf-Theoretic Approach

Arne Wolf^{1,2} and Anthea Monod¹(✉)

¹ Imperial College London, London SW7 2AZ, UK
{a.wolf22,a.monod}@imperial.ac.uk

² London School of Geometry and Number Theory, London WC1E 6BT, UK

Abstract. We propose a model for network community detection using topological data analysis, a branch of modern data science that leverages theory from algebraic topology to statistical analysis and machine learning. Specifically, we use cellular sheaves, which relate local to global properties of various algebraic topological constructions, to propose three new algorithms for vertex clustering over networks to detect communities. We apply our algorithms to real social network data in numerical experiments and obtain near optimal results in terms of modularity. Our work is the first implementation of sheaves on real social network data and provides a solid proof-of-concept for future work using sheaves as tools to study complex systems captured by networks and simplicial complexes.

Keywords: Cellular sheaves · community detection · modularity · opinion dynamics · topological data analysis

1 Introduction

Networks are used to describe, study, and understand complex systems in many scientific disciplines. One of the most important features in complex systems that networks are able to capture is the presence of *communities*. In networks, communities can be seen as partitioning a graph into clusters, which are subsets of vertices with many edges connecting the vertices within the subset, and comparatively fewer edges connecting to different subsets in the rest of the network. These clusters or communities can be considered as relatively independent components of a graph. The problem of *community detection* is to locate those clusters in networks which are more strongly connected than the whole network is, on average. Community detection is a challenging and active area of research in network science: a key aspect that makes the problem difficult is that there is no single, universally accepted definition of a community within a network and it is largely dependent on context or the specific system being studied.

Topological data analysis (TDA) is a recently-emerged approach to data science that uses principles from pure mathematics to extract meaningful information from large and complex datasets that may not possess a rigorous metric

or vector space structure which is often required in classical data analysis. In our work, we focus on *sheaves*, which are a tool that relates local to global properties of various constructions in algebraic topology and have been used to reinterpret and generalize many significant results in classical geometry and algebraic topology. Sheaves allow for information to be assigned to subsets of topological spaces; when the topological space is a network and the information is vector space-valued, a computational framework for sheaves becomes available similar to that of *persistent homology*, which is a well-developed and the most widely-used tool in TDA; see [2] for complete details.

In this paper, we propose a topological approach to community detection based on sheaves. Specifically, we show that sheaves may be used to rigorously model the problem of community detection on a network and propose three novel, sheaf-based community detection algorithms. We test and compare their performance with numerical experiments on a real-world benchmarking dataset and show that we are able to attain near optimal community detection results.

Related Work. Particularly relevant to our work, sheaves have been previously used to model various dynamics of opinions over social networks, including bounded confidence models, stubbornness, and the formation of lies [3]. In a similar spirit, propagation of gossip has also been theoretically modeled using sheaves [5]. However, it is important to note that no implementations nor applications to real data exist of these sheaf social network models. In a non-topological setting, the formation of opinion clusters has been investigated in the bounded confidence model [4]; another contrast to our work is that they use dynamics with discrete time steps and sharp confidence bounds. Very recently, the only other TDA approach to community detection that we are aware of was proposed [6], however, it is based on persistent homology, rather than sheaves, as in our work.

2 Background: Sheaves and Social Networks

In this section, we define sheaves and present sheaf-theoretic notions in the context of social networks that will be used in our proposed community detection algorithms.

2.1 Sheaves and Sheaf Cohomology

Sheaves assign data to open subsets of topological spaces X in a consistent manner. This compatibility is what enables them to relate *local* (i.e., the pure data) and *global* (i.e., having an assignment of compatible data) properties of X . *Cellular* sheaves are a special case where X is a cellular complex (which is a generalization of a simplicial complex) and the data are vector spaces. In this paper, we will further restrict to the case that the cellular complex is a graph $G = (V, E)$; we consider undirected graphs without loops or multiple edges. We write edges as ordered tuples $e = (v_1, v_2)$, ordered arbitrarily. Under these

restrictions to graphs, we now outline how important concepts from graph theory generalize to the topological setting of sheaves which will be relevant to our work further on.

The graph G becomes a topological space when exploiting the fact that the *subface relation* allows G to be viewed as a partially ordered set and to be equipped with the Alexandrov topology. A sheaf on G turns out to be uniquely determined by the following data [2].

Definition 1. A (cellular) sheaf \mathcal{F} assigns finite-dimensional real vector spaces, called stalks, $\mathcal{F}(v)$ to each vertex and $\mathcal{F}(e)$ to each edge of a graph, and a linear map $\mathcal{F}_{v \subset e} : \mathcal{F}(v) \rightarrow \mathcal{F}(e)$ (called a restriction) to each incidence $v \subset e$.

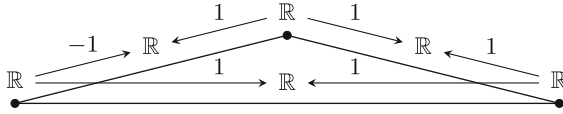


Fig. 1. Example of a cellular sheaf

Figure 1 illustrates a cellular sheaf. Other examples that can be defined for any graph G are the *constant sheaves* $\underline{\mathbb{R}}^n$ for $n \in \mathbb{N}$, where all stalks are \mathbb{R}^n and all restrictions the identity map.

Algebraic topology is a field of pure mathematics that uses abstract algebra to study topological spaces; specifically, it defines algebraic ways of counting properties of topological spaces that are left unaltered under continuous deformations of the topological space, such as stretching or compressing. Such properties are referred to as *invariants*; *cohomology groups* are examples of such invariants. Sheaves over topological spaces give rise to *sheaf cohomology groups*. In our setting, these are obtained from a collection of vector spaces and maps between them, called *the cochain complex* [7, Theorem 1.4.2]

$$0 \longrightarrow C^0(G, \mathcal{F}) \xrightarrow{\delta} C^1(G, \mathcal{F}) \longrightarrow 0 \longrightarrow \dots$$

with cochain groups

$$C^0(G, \mathcal{F}) := \bigoplus_{v \in V} \mathcal{F}(v), \quad C^1(G, \mathcal{F}) := \bigoplus_{e \in E} \mathcal{F}(e).$$

Here, the linear *coboundary map* $\delta : C^0(G, \mathcal{F}) \rightarrow C^1(G, \mathcal{F})$ is defined by acting linearly on stalks. On $\mathcal{F}(v)$, it acts according to

$$x_v \mapsto \sum_{e=(v_k, v)} \mathcal{F}_{v \subset e} x_v - \sum_{e=(v, v_l)} \mathcal{F}_{v \subset e} x_v.$$

The sheaf cohomology groups of interest are then

$$H^0(G, \mathcal{F}) = \ker(\delta) \quad H^1(G, \mathcal{F}) = C^1(G, \mathcal{F}) / \text{im}(\delta) = \text{coker}(\delta).$$

Example 1. The graph in Fig. 1 has trivial cohomology with respect to the illustrated sheaf, even though the classical (e.g., simplicial) cohomology is nontrivial. This shows that these two cohomology theories need not agree.

Sheaf cohomology is in fact a generalization of classical cohomology, because for the constant sheaf \mathbb{R}^1 , the sheaf cohomology yields precisely the cellular cohomology of G . In this case, $\delta = B^\top$ is the transpose of the *signed incidence matrix* B which is defined by

$$B_{v,e} = \begin{cases} 1 & \exists w \in V : e = (v, w), \\ -1 & \exists w \in V : e = (w, v), \\ 0 & \text{otherwise.} \end{cases}$$

Recall that the classical *graph Laplacian* can be obtained as $L = BB^\top$. This notion may be generalized to obtain the following definition.

Definition 2. For a cellular sheaf \mathcal{F} over G , the sheaf Laplacian is $L_{\mathcal{F}} := \delta^\top \delta$.

It can be shown that neither sheaf cohomology nor the sheaf Laplacian depend on the initially chosen orientation.

2.2 Discourse Sheaves and Opinion Dynamics

By considering social networks G where persons are modeled by vertices and acquaintanceship is modeled by connections, the distributions of opinions may be expressed by a *discourse sheaf* \mathcal{F} [3]; see Fig. 2 for an illustration.

Each person v is assigned an *opinion space* $\mathbb{R}^{n_v} = \mathcal{F}(v)$ and *opinion* $x_v \in \mathcal{F}(v)$. A basis of $\mathcal{F}(v)$ can be seen as collection of basic topics that v cares about and the component of x_v in a basis direction expresses

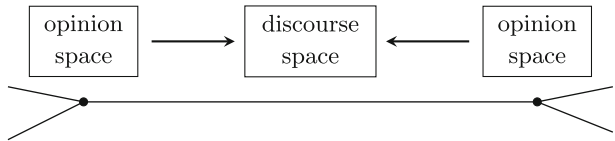


Fig. 2. Sketch of a discourse sheaf

the opinion about that topic (e.g., how much v supports a certain politician).

Edges stand for discourse about topics in $\mathbb{R}^{n_e} = \mathcal{F}(e)$. There need not be a relation between bases of different stalks. However, each person v projects their opinion of the discussed topics on e via $\mathcal{F}_{v \subset e}$. There is *consensus* along $e = (u, v)$ if $\mathcal{F}_{v \subset e} x_v = \mathcal{F}_{u \subset e} x_u$.

In this framework, several models have been proposed to describe how opinions expressed by such a sheaf evolve over time [3]. The basic model assumes that everyone changes their opinion in order to minimize the difference to the average opinion of their friends, i.e.,

$$\frac{d}{dt} x_v(t) = \sum_{v \overset{e}{\sim} u} \mathcal{F}_{v \subset e}^\top (\mathcal{F}_{u \subset e} x_u - \mathcal{F}_{v \subset e} x_v)$$

Here $v \overset{e}{\sim} u$ denotes that v and u share a common edge e . Combining all opinions to one vector x , this can be written as $\frac{d}{dt}x(t) = -L_{\mathcal{F}}x$. Solutions converge exponentially to consensus on all edges ([3], Theorem 4.1).

The more realistic *bounded confidence model* assumes that the influence of a friend’s opinion on v decreases if their opinions differ too much. The decay is expressed by a monotonically decreasing “bump” function $\phi : [0, \infty) \rightarrow [0, 1]$ that vanishes precisely whenever some threshold D is surpassed, as shown in Fig. 6. The modified dynamics are described by

$$\frac{d}{dt}x_v(t) = \sum_{v \overset{e}{\sim} u} \phi(\|\mathcal{F}_{u \subset e} x_u - \mathcal{F}_{v \subset e} x_v\|) \mathcal{F}_{v \subset e}^{\top} (\mathcal{F}_{u \subset e} x_u - \mathcal{F}_{v \subset e} x_v). \quad (1)$$

Configurations are stable if on each edge the difference is zero or at least D . In such a stable configuration, disregarding all edges without consensus and considering the connected components of the remaining graph provides a partition of the vertex set V (persons in the social network).

3 Methods and Experimental Design

In this section, we outline our proposed sheaf-theoretic algorithms used for community detection. We propose two algorithms that generate partitions of a given graph at random and one deterministic algorithm and evaluate their ability to detect communities. For the sake of comparability, we consider partitions of the full vertex set of a fixed graph (see Fig. 3 for an example of such a partition) and evaluate how well they represent the community structure of the graph. The common evaluation measure that we will use in this work is *modularity*, which is computed in an assessment step to determine whether the division into communities is “relatively good.”

Definition 3. Let $G = (V, E)$ be a graph and V be subdivided into N subsets $\{V_c \mid c = 1, \dots, N\}$, with E_c being the set of edges between nodes from V_c . Then the modularity Q is given by

$$Q := \sum_{c=1}^N \left[\frac{|E_c|}{|E|} - \left(\frac{\sum_{v \in V_c} \deg(v)}{2|E|} \right)^2 \right] \in [-1, 1]. \quad (2)$$

Intuitively, the quantity Q captures how many more edges than the average are within the subsets V_c , so high modularity indicates a good partitioning.

3.1 Detecting Communities with Constant Sheaves

Our first algorithm considers constant sheaves and models dynamics after the sheaf-theoretic bounded confidence model previously described in Sect. 2.2.

In Step 2 of Algorithm 1, we have included a stopping criterion for time efficiency: If the evolution does not converge after 1000 time units, the calculation is aborted. In Sect. 3.2, we explain why this is necessary. We keep track of the number of abortions and take them into account in our reported results below, when providing uncertainties.

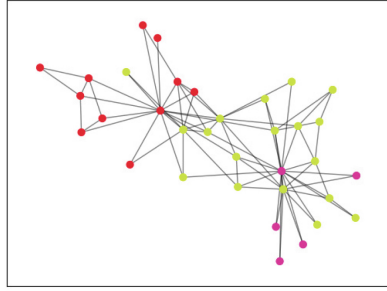


Fig. 3. “Karate club” graph partitioned into three communities.

Algorithm 1: Community Detection with Constant Sheaves \mathbb{R}^n

Input: dimension parameter n , parameter for diameter d

- 1 Initialize by picking a random opinion vector for each vertex stalk uniformly from $B(0, \frac{d}{2})$
- 2 Evolve the system using Eq. (1) until for no edge $e = (u, v)$, the difference $\|x_u - x_v\| \in (0.0033, 1)$; abort if that does not happen within 1000 time units
- 3 Obtain primary partition by grouping neighbors w, w' together if and only if $\|x_w - x_{w'}\| \leq 0.0033$
- 4 **if** we find a single-vertex community v **then**
- 5 add the vertex to that adjacent community C , which has maximal $|E|k_C - \deg(v) \cdot \sum_{w \in C} \deg(w)$, where k_C is the number of neighbors of v belonging to C

Output: The obtained partition

We impose the positive value of 0.0033 for the computed difference in Steps 2 and 3 as another stopping criterion and trade-off between precision and time-efficiency.

Step 4 resolves single-vertex clusters by adding the vertex v to the cluster of that neighbor of v which is optimal in the sense of modularity. Proposition 1 justifies this choice by showing that each of these moves increases modularity.

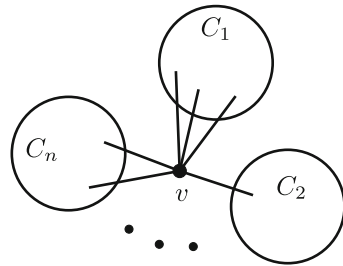


Fig. 4. Sketch of a possible situation in the proof of Proposition 1 with $k_1 = 3, k_2 = 1, k_n = 2$

Proposition 1. *Single vertex clusters can be removed in a way that increases modularity.*

The removal of single vertex clusters implemented in Algorithm 1 is optimal in the sense of modularity.

Proof. Consider a vertex v , the clusters C_1, \dots, C_n to which its neighbors belong, and let k_i be the number of neighbors of v that are in C_i , see Fig. 4. Adding v to C_i increases the modularity of the given partition by

$$\Delta_i := \frac{k_i}{|E|} - \frac{2 \deg(v) \cdot \sum_{w \in C_i} \deg(w)}{4|E|^2}.$$

Notice that the value that we maximize in Step 5 is $2|E| \Delta_i$ (for the cluster C_i considered in Step 5). We must show that at least for one i , Δ_i is positive. Observe that

$$\begin{aligned} 4|E|^2 \sum_{i=1}^n \Delta_i &= \sum_{i=1}^n 4|E|k_i - 2 \deg(v) \sum_{i=1}^n \sum_{w \in C_i} \deg(w) \\ &> 4|E| \deg(v) - 2 \deg(v) \cdot 2|E| = 0, \end{aligned}$$

because $\sum_{i=1}^n \sum_{w \in C_i} \deg(w)$ is bounded from above by the sum of all degrees of vertices that are not v which is $2|E| - \deg(v)$. This completes the proof.

3.2 Convergence of Algorithm 1: Community Detection with Constant Sheaves

The following example shows why we need the abortion criterion in Step 2 of Algorithm 1.

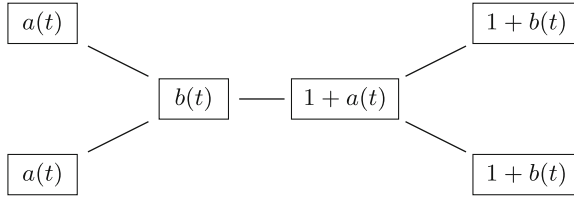


Fig. 5. Example of a network in which Algorithm 1 is aborted in Step 2

Example 2. Consider the graph shown in Fig. 5 and the constant sheaf \mathbb{R} on that graph. Let

$$\phi(x) = \begin{cases} 1 - x & 0 \leq x \leq 1 \\ 0 & x \geq 1 \end{cases}.$$

Initialize with values a_0 and b_0 such that $1 + a_0 > b_0 > a_0$, making sure that none of the connections drawn in Fig. 5 is ignored. Applying Eq. (1), we check that all vertices with value $a(t)$ and $1 + a(t)$ as well as all with $b(t)$ and $1 + b(t)$

follow the same derivative. Therefore, the opinions in the network will remain in the pattern shown in Fig. 5 and the values $a(t)$ and $b(t)$ evolve according to

$$\frac{da(t)}{dt} = (b(t) - a(t))(1 + a(t) - b(t)) = -\frac{db(t)}{dt}.$$

This yields

$$\begin{aligned} a(t) &= \frac{1}{2} \left(a_0 + b_0 - \frac{1}{e^{2t+c} + 1} \right) \\ b(t) &= \frac{1}{2} \left(a_0 + b_0 + \frac{1}{e^{2t+c} + 1} \right) \end{aligned}$$

with $c = \ln\left(\frac{1}{b_0 - a_0} - 1\right)$. In particular, $1 + a(t) - b(t) = 1 - \frac{1}{e^{2t+c} + 1} < 1$, but it converges to one. Thus, the difference over the edge in the middle of Fig. 5, between $b(t)$ and $1 + a(t)$ will always end up in the interval $(0.0033, 1)$ and Algorithm 1 will be aborted in Step 2.

Even though there are configurations without convergence, in our experiments we observed a convergence within 1000 time steps for more than 95% of starting configurations for any choice of parameters.

3.3 Detecting Communities with a Non-constant Sheaf

We now consider a second community detection algorithm that uses the non-constant sheaf defined by setting $\mathcal{F}(v) = \mathbb{R}^{\deg(v)}$ (we think of it as one copy of \mathbb{R} for every edge that uses v), $\mathcal{F}(e) = \mathbb{R}$, and restriction maps being the projections onto the corresponding edges.

At the vertices, there is no interaction between the directions corresponding to different edges, so each direction can be treated separately. If for an edge $e = (v, w)$, the difference between the initial components $x_{v,e}$ and $x_{w,e}$ is less than D , edge e will “survive” the evolution and otherwise not. Therefore, picking initial values at v uniformly from $[-\frac{d}{2}, \frac{d}{2}]^{\deg(v)}$ amounts to keeping each edge with the same probability p (which depends only on d) and ignoring it otherwise. We thus obtain Algorithm 2.

Algorithm 2: Community Detection with a Non-constant Sheaf

Input: probability parameter p

- 1 Obtain primary partition by grouping neighbors w, w' together with probability p
- 2 Remove single-vertex clusters as in Algorithm 1, Step 4

Output: The obtained partition

3.4 Deterministic Sheaf Community Detection

A more general version of the bounded confidence model allows different functions ϕ_e for different edges. In the case of the non-constant sheaf, this means that the probability p can depend on the edge and the local structure of the graph.

With the idea in mind that we want to retain edges if the vertices belong to the same community, it should be more likely for $e = (u, v)$ to be retained if the number of common neighbors of u and v , denoted $N_{u,v}$, is not too small compared to the total number of neighbors of u and v . In an extreme case, this likelihood is either zero or one, and we arrive at Algorithm 3.

Algorithm 3: Deterministic Community Detection

Input: two parameters $a \in [0, 1]$ and $b \in \mathbb{R}$

- 1 Obtain primary partition by grouping the two ends u, v of an edge e together if $a \cdot (\deg(u) + \deg(v)) < b + N_{u,v}$
- 2 Remove single-vertex clusters as in Algorithm 1, Step 4

Output: The obtained partition

3.5 Experimental Setup

We tested the performance of Algorithm 1 by considering its dependence on d for $n = 1$ and the different bump functions ϕ shown in Fig. 6, as well as for $n \in \{2, 3, 5, 10\}$ and ϕ_1 . Note that for sake of comparability we fixed $D = 1$ to be the threshold for ignoring. On the interval $[0, 1)$, the ϕ_i are given by:

$$\begin{aligned} \phi_1(x) &= 1 - x & \phi_2(x) &= 1 - x^2 \\ \phi_3(x) &= (1 - x)^2 & \phi_4(x) &= 1 - x - \sin(2\pi x)/7 \end{aligned}$$

Algorithm 2 was run for various values of p and Algorithm 3 with different values of a and b .

All tests were performed on Zachary’s karate club graph G , shown in Fig. 3, which represents a social network of a karate club studied by sociologist Wayne Zachary from 1970 to 1972 [8]. The network captures 34 members of the karate club and includes links between pairs of members who interacted outside the club. The maximal possible modularity of a partition of G for this graph is $Q_{\max}(G) \approx 0.42$ [1]. All experiments were implemented in Python and the code is freely and publicly available at https://github.com/ArneWolf/Bd_confidence_communities.

4 Experimental Results

We report the results of our numerical experiments for our three proposed sheaf-theoretic community detection algorithms from Sect. 3. We compare the performance in terms of average number of clusters and modularity (Definition 3).

Algorithm 1: The Constant Sheaves. To justify our stopping criterion of 0.0033 in Step 2 of Algorithm 1 and estimate the error caused, we compared the partitions

to those obtained for running the evolution longer so that for every edge $\|x_u - x_v\| \notin (0.001, 1)$ instead. The more precise calculations took, on average, a factor of 100 longer. Out of 2930 runs we performed, the first stopping criterion applied 175 times and for the remaining 2755 runs, the two obtained partitions agreed. We thus concluded that 0.0033 is small enough to not cause a significant error.

Each parameter combination was simulated $N = 1000$ times. Figure 8 shows the average values of X (number of clusters, modularity) and an error bar with radius

$$X_{\text{error}} = \sigma_X + \frac{\# \text{ early stoppings}}{N} \bar{X},$$

where \bar{X} denotes the average and σ the standard deviation.

We immediately see that a larger width d of the distribution from which the initial opinions are picked results in more clusters. Neither the number of clusters nor the modularity depends significantly on the choice of ϕ .

In the case of smaller d , however, ϕ_3 tends to create more clusters than the others. Intuitively, this behavior can be explained by comparing ϕ_3 to linearly decaying bump functions: it is more similar to a bump function that decays quicker than ϕ_1 , than to ϕ_1 . Quicker decay means a smaller threshold $D < 1$. That in turn corresponds to having $D = 1$ and a bigger d . Thus, the properties of ϕ_3 tend to make d appear larger.

For large d , the influence of removing singleton clusters (compared to the primary partition) grows. This explains why the differences in the number of clusters as well as modularity for different choices

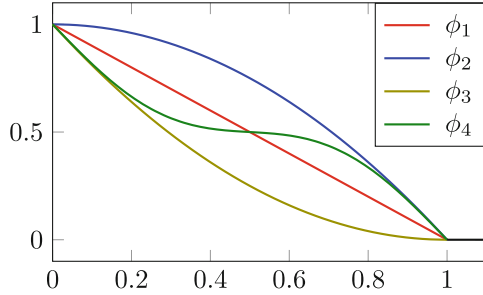


Fig. 6. The different functions ϕ

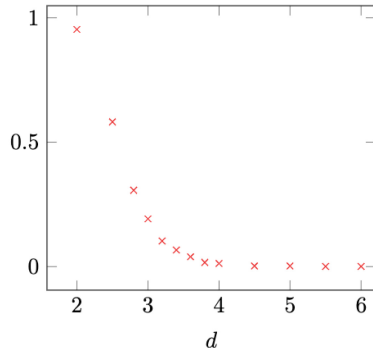


Fig. 7. Likelihood of most likely partition for different diameters d of the initial distribution

of ϕ and d become smaller. Looking at $p = 0$ in Fig. 10 shows that resolving singleton clusters from a collection of only singleton clusters yields a modularity of about 0.191. This is one explanation for having higher modularities for larger d . Another explanation is that small d are likely to result in one big cluster containing all the points. This trivial partition has modularity 0. Figure 7 shows the likelihood of the most likely partition for ϕ_1 and $n = 1$. For $d \leq 4$, whenever there is a partition that is significantly the most likely one, this is the trivial partition. The same qualitative behavior is observed for other ϕ and n .

If d is large enough, the number of clusters increases as n increases. This is significant for $n = 10$, but the tendency can be observed in Fig. 9 for smaller n as well. This can be explained by larger n corresponding to more topics of discussion, which in turn increases the chances that persons can disagree. The modularity for small d seems to depend on n non-monotonously: $n = 10$ gives the highest values, followed by $n = 1$ and $n = 2$. If d is large, the differences become insignificant.

Algorithm 2: The Non-Constant Sheaf. For Algorithm 2, Fig. 10 shows the expected decrease of the number of clusters with increasing p . The modularity has a maximum of about 0.26 near $p = 0.12$. For smaller $p \rightarrow 0$, the modularity becomes 0.191 and for large p it goes to zero due to the dominance of the trivial partition.

Algorithm 3: The Deterministic Algorithm. Due to the affine linear condition in Step 1 of Algorithm 3, the modularity and number of clusters change along affine lines displayed in Fig. 11. The deterministic algorithm reaches a maximal modularity of about 0.407, which comes close to $Q_{\max} = 0.42$. However, the portion of the parameter space that reaches this value is small—the other algorithms had broader ranges of maxima for their modularities. The partition yielding $Q = 0.407$ consists of four clusters, just as the partition obtaining Q_{\max} .

5 Discussion

In this work, we showed that sheaves are a viable algebraic topological tool to model the problem of community detection on networks. We proposed three different algorithms, two of which had random initializations and were based on constant sheaves and a non-constant sheaf, and a third deterministic version that allows for different bump functions for different edges of the graph. The deterministic sheaf-theoretic community detection algorithm, in particular, performed well in terms of modularity and achieved values near the maximal modularity value. Ours is the first work to computationally implement sheaves on real-world social network data and the first use of sheaves in the problem of community detection on networks. Moreover, our work provides a proof-of-concept for future work adapting the potential of cellular sheaves to studying complex systems captured by networks and simplicial complexes, in general.

Directions for future work involve combining various notions from the different algorithms to, for instance, allow bump functions in Algorithm 1 or the

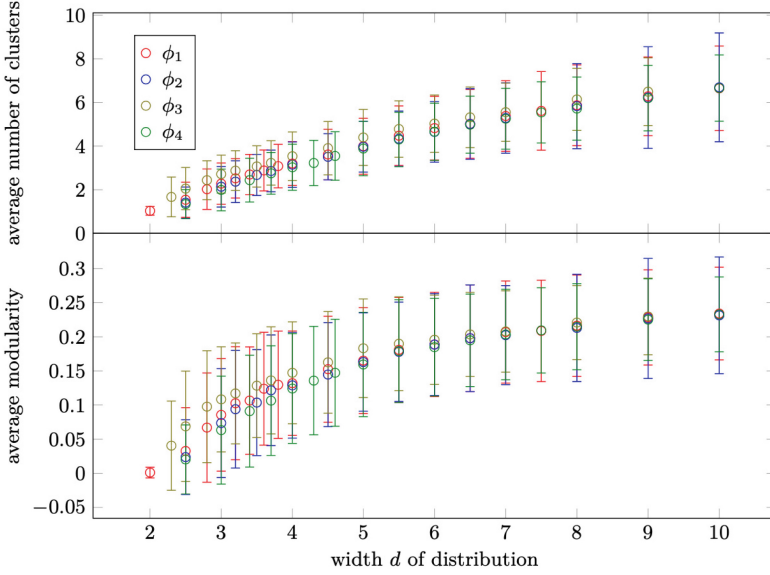


Fig. 8. Average number of clusters and modularity for $n = 1$, different bump functions ϕ and different diameters of the initial opinion distributions

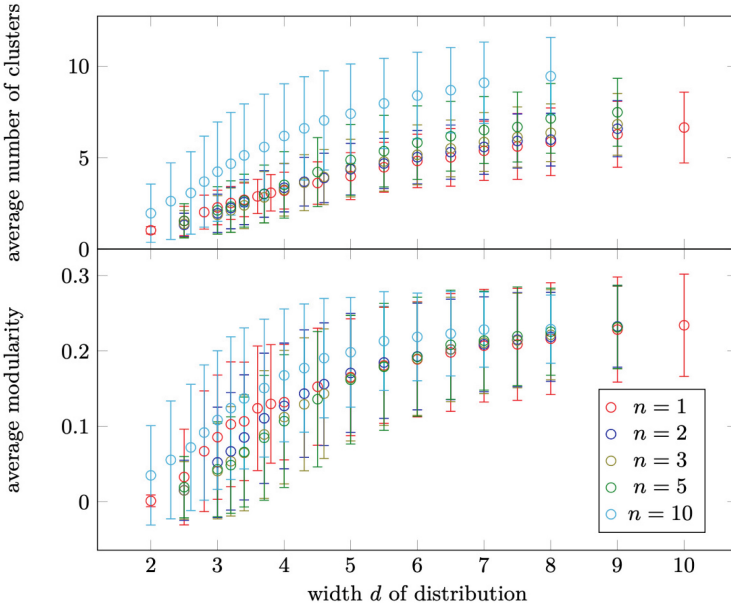


Fig. 9. Average number of clusters and modularity for different dimensions n , bump function ϕ_1 and different diameters of the initial opinion distributions

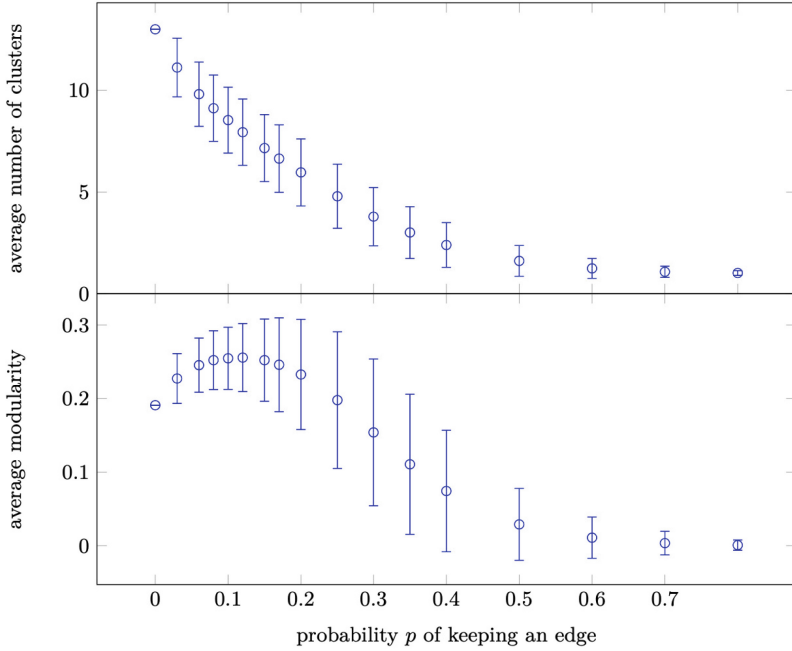


Fig. 10. Average number of clusters and modularity for the non-constant sheaf algorithm for different probabilities p of keeping an edge

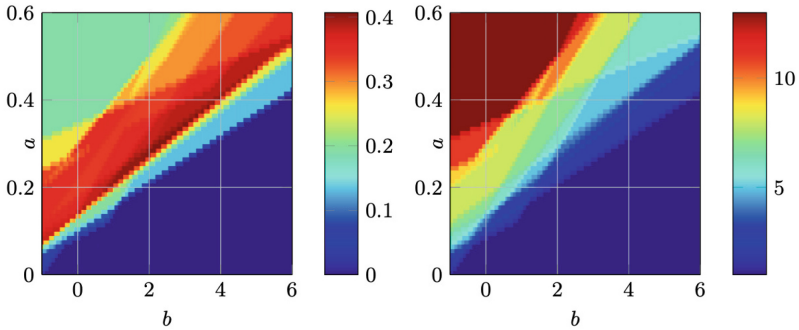


Fig. 11. Modularities (left) and numbers of clusters (right) for several parameter combinations for the deterministic Algorithm 3

probability p in Algorithm 2 to depend on the edge, which may potentially improve performance. An optimization procedure may also be proposed to find optimal parameters, and thus, optimal partitions for a network.

Acknowledgments. A.W. is funded by a London School of Geometry and Number Theory–Imperial College London PhD studentship, which is supported by the Engineering and Physical Sciences Research Council [EP/S021590/1].

References

1. Costa, A.: Some remarks on modularity density. [arXiv:1409.4063](#) (2014)
2. Curry, J.M.: Sheaves, cosheaves and applications. University of Pennsylvania (2014)
3. Hansen, J., Ghrist, R.: Opinion dynamics on discourse sheaves (2020)
4. Hegselmann, R., Krause, U.: Consensus and fragmentation of opinions with a focus on bounded confidence. *Am. Math. Mon.* **126**(8), 700–716 (2019)
5. Riess, H., Ghrist, R.: Diffusion of information on networked lattices by gossip (2022)
6. Schindler, D.J., Barahona, M.: Persistent homology of the multiscale clustering filtration. arXiv preprint [arXiv:2305.04281](#) (2023)
7. Shepard, A.D.: A cellular description of the derived category of a stratified space. Brown University (1985)
8. Zachary, W.W.: An information flow model for conflict and fission in small groups. *J. Anthropol. Res.* **33**(4), 452–473 (1977)



Does Isolating High-Modularity Communities Prevent Cascading Failure?

Stephen Eubank^(✉)

Biocomplexity Institute, University of Virginia, Charlottesville, VA, USA
eubank@virginia.edu

<https://biocomplexity.virginia.edu/person/stephen-eubank>

Abstract. Communities are often defined as sets of nodes that are more densely connected to each other than to those outside the community, i.e., high-modularity partitions. It seems obvious that isolating high-modularity communities is a good way to prevent the spread of cascading failures. Here we develop a heuristic approach informed by Moore-Shannon network reliability that focuses on dynamics rather than topology. It defines communities directly in terms of the size of cascades they allow. We demonstrate that isolating communities defined this way may control cascading failure better. Moreover, this approach is sensitive to the values of dynamical parameters and allows for problem-specific constraints such as cost.

Keywords: cascading failure · dynamics · reliability

1 Introduction

The notion of community in a network is often introduced to characterize an aspect of network structure that may place important constraints on dynamical phenomena. Hence the assumption that the “proper” definition of community in Zachary’s karate network [22] would match its eventual partition in two. Among the most important dynamical phenomena in networked systems are cascading failures, e.g., the spread of infectious disease across a contact network or the spread of power outages across an electrical power distribution network. The inverse of Granovetter’s “strength of weak ties” is the fragility of inter-community cascade: although it may be difficult to prevent cascades *within* a community, by cutting the edges between communities we can prevent system-wide cascades. This motivates defining communities as nodes that are more densely connected to each other than to those outside the community [17].

Under this definition, finding communities reduces to maximizing modularity. This is a clustering problem that has been extensively studied in the computer and network science literatures [1, 7, 11, 13, 18, 21]. In particular, a k-means clustering of eigenvectors of the adjacency matrix yields partitions with high modularity and low cost in undirected networks and can be extended to directed,

weighted networks. However, modularity only captures local coupling, not long-range interaction. Moreover, because it is independent of any dynamics operating over the network, there is no *a priori* reason that it should be relevant for cascading failures.

We might consider extending the definition of community by replacing “density of edges” with “strength of interactions” as determined by some dynamics. We characterize the strength of an interaction between two nodes (not necessarily connected by an edge) as the *influence* of node A on node B under dynamics D . Specifically, the influence is the probability that a disturbance to A 's state – e.g., a “failure” – will propagate to B . For a deterministic system, the influence is either 0 or 1 for every pair of nodes; for stochastic systems it is more nuanced. This is true whether the stochasticity arises from the dynamics D , as for infectious disease transmission, or from an imperfect knowledge of the system's state, such as electric power networks which may have hidden failures in the control system. Influence is not necessarily symmetric, and it depends in complicated ways on all paths, not just the shortest ones, connecting A to B .

A hallmark of complex networks is that, just like a lens, they can focus propagating disturbances so that, for example, two distant nodes' states may be much more closely related than any of those on paths joining them. What does a community look like in such networks under this definition? Unfortunately, it is not necessarily consistent with our intuitive notions of community:

1. Connectivity: if A strongly influences B , but only indirectly through a large set of weakly coupled vertices, should A and B be placed in the same community? What about the weakly coupled intermediate vertices?
2. Direction: should communities be strongly connected components? Conversely, should all the nodes of a strongly connected component be placed in a single community?
3. Reflexivity: if B is in A 's community, should we require that A be in B 's?
4. Transitivity: if B is an element of A 's community and C an element of B 's, should we require that C is also an element of A 's?
5. Scale and hierarchy: should there be different communities defined at different scales, and, if so, should they be nested?

Rather than answer all these questions to find communities that are, after all, only plausible candidates for preventing system-wide cascades, we present here a way to find cuts that prevent cascades directly and define communities as connected components of the subgraph induced by the cuts. The analysis does not depend on the spectrum of the adjacency matrix, so it can be applied immediately to directed networks. Moreover, it incorporates dynamics essentially, so it is sensitive to dynamical parameters that are not captured in a purely static, topological statistic. It is motivated by Moore-Shannon Network Reliability (MSNR) [15], but it is not simply minimizing inter-community influence, which is essentially the two-point reliability. It relies on being able to find highly probable failure cascades. For comparison to modularity based communities, we only need good cuts, and we do not claim that MSNR finds the optimal cuts.

In this paper, we compare the size of cascading failures to both the modularity and the inter-community influence on partitions defined by the directed Louvain (DL) algorithm [3, 12] and MSNR. DL is designed to produce high-modularity partitions and does so; MSNR is designed to produce small cascades and does so. On one hand, for a set of planted, ℓ -partition (PLP) graphs, we show that the two methods produce comparable partitions, and that all three metrics are strongly associated with the cost of the edge cut induced by the partitions, and are thus themselves closely related. On the other hand, for a set of directed, weighted commodity trade networks with edge costs, there is no obvious relationship among the three metrics for either method.

If the goal of network structural analysis is to determine a good strategy for controlling dynamical phenomena such as the maximum size of cascading failure on a network, the MSNR approach produces better communities. In complex networks, modularity is not sufficiently relevant to the task. We expect that for other goals that can be formulated in terms of network reliability, such as isolating a specific set of nodes, analogous direct methods will also be superior.

2 Methods

We simulate epidemics of infectious “failure” with varying overall transmissibility over a collection of networks, both artificial and real, and compare the distributions of total number infected to simulated distributions on subgraphs of isolated communities determined by the (directed, weighted) Louvain and MSNR algorithms. We consider whether the reduction in size of the cascade is related to the modularity or inter-community influence of either collection of communities. Here we describe the MSNR algorithm, the networks, the simulated dynamics, and estimation of inter-community influence.

2.1 MSNR

We have previously [14] constructed communities using a greedy algorithm that iteratively removes edges that contribute most to the probability of a large outbreak. This probability is a kind of Moore-Shannon reliability and is thus provably hard to evaluate [19, 20]. We have used two approximations: simulation and perturbative approximation [2, 4]. In general, we find that the difference in contributions between edges is so small it requires enormous numbers of simulations to estimate confidently. Perturbative approximation can, in principle, resolve these small differences, but also requires sampling from a universe that is combinatorially large in the number of edges E . If S is the number of samples, the k th nonzero term in the perturbation series captures interactions among k samples and requires S^k computations. It is not yet clear how the quality of approximation depends on the sample size, but it does not seem likely the approach will scale to extremely large networks.

We thus turn to the following heuristic. A cascade of failures resulting from the failure of a single node v can be thought of as a subgraph G' of the network

that includes only the failed nodes and the edges among them. Assuming that failure must spread along edges, the conditional probability of the cascade given that v fails is the probability that the dynamics “picks” any of the combinatorially many minimal subsets of edges required to generate failures in exactly the nodes of G' . For complex contagions, these minimal subsets may be complicated structures like k -cores; but for the dynamics we are concerned with here, described in Sect. 2.3, they are spanning trees. Hence the probability of a cascade of size N that includes a particular node v is at least as large as the probability of the most probable tree with N vertices that includes v . This simple heuristic leads to a deterministic algorithm that scales linearly with V and with CE/V , where C is the maximum tolerable size of a cascade, because the bulk of the work is finding V paths of length C .

Our approach has two stages:

1. identify, for each node v , a set of nodes S_v that is highly probable to be involved in any cascade that is undesirably large involving v .
2. construct a partition of the node set that is reasonably consistent with the collection S_v and their probabilities and satisfies our intuitive notions of community, i.e., our answers to the questions posed in the Introduction.

The first stage is implemented as a breadth-first greedy optimization from v (both upstream and downstream if the network is directed) that chooses the most probable node on the frontier at each step. The computation of downstream probabilities can take into account multiple paths from v , although we see no simple way to do this for upstream nodes. The second stage is left intentionally vague to allow construction of communities meeting different criteria. It is the same problem as identifying communities in a friendship network where each person has nominated N friends and one must decide how to represent these non-symmetric, non-transitive lists as a collection of communities.

For the results described here, we implemented the second stage as follows. Starting from a randomly chosen node v , we consider the intersection of the sets S_w for each w in S_v . If this intersection contains more than the node v , we label it a community; otherwise, we choose a node that has not yet been chosen or assigned a community and repeat the process. We clean up the remaining isolated vertices by including them in the most likely upstream or downstream community with fewer than C nodes, if any. In a final pass, we coalesce any communities that are connected in the original network if their sizes allow.

2.2 Networks

Planted ℓ -Partition (PLP) Networks. A PLP network is equivalent to an instance of a stochastic block model. It is a random graph with known community structure. We have planted four “communities”, each with 25 nodes. The mean degree of each node is 16, and on average, k of each node’s neighbors are in a different community. As k increases from one, the planted communities become less obvious until, at $k = 8$, they are completely invisible. We created

ten uniformly weighted, undirected PLP instances for each value of k from 1 to 7.

PLP networks are designed for testing algorithms to optimize modularity. Importantly, algorithms can be evaluated against the correct partition instead of each other. The free parameter k can be used conveniently to adjust the optimal modularity and the difficulty of the optimization problem. Their structure is highly artificial, however, and as we show below, results on PLP may not generalize to more complex, real networks.

Commodity Networks. We consider weighted, directed commodity trade networks described in [14, 16]. These are country-to-country networks from the Food and Agriculture Organization’s Trade Matrix database [6] and U.S. domestic food networks from the Freight Analysis Framework [5]. Weights on each edge represent the volume of commodity shipped from one region to another. We interpret these weights as both costs and transmission probabilities, but in different ways. We define the fractional cost associated with removing an edge as its weight relative to the total volume of trade. However, we treat the weights as Poisson process rates in the dynamics, as described in Sect. 2.3.

2.3 Dynamics

Characterizing communities with respect to dynamical processes is an emerging body of work. Ghosh et al. [8] define a generalized Laplacian matrix that captures a class of linear dynamical processes. They introduce the notion of generalized conductance to measure the quality of communities with respect to the dynamical process. However, their work is limited to undirected networks. In another line of work, Zhang et al. [23] consider the problem of discovering clusters of nodes that have similar roles in a dynamical process (e.g., influential nodes or bridges).

Propagation in Random Media. A natural metric of interaction strength in a dynamical system is the response of each node’s state to a perturbation of another node’s state. However, in many systems there is no natural metric on the states themselves. An alternative is the change in the probability distribution of each element’s state when the system is perturbed. We call the set of all node states at time t the system’s *configuration* $C(t)$. The dynamics transform a probability distribution over system configurations $P_C(t)$ to $P_C(t+1)$. Perturbing the system at time t means introducing a different probability distribution $P_C(t) \rightarrow P'_C(t)$. This, in turn, leads to a different probability distribution at $t+1$: $P_C(t+1) \rightarrow P'_C(t+1)$. Our definition of community flexibly allows for restrictions on both the kinds of perturbations and the features of the difference between P and P' that are of interest. Here we restrict the initial perturbation to a change of state in a single node and use as a metric the difference in the range of the top two quartiles of P and P' .

SIR. In some cases, we can model the spread of a perturbation as an *SIR* process on a directed, weighted network. That is, we consider a node whose state has not been perturbed to be “Susceptible”; a node whose state has been perturbed will be considered “Infectious” immediately after the perturbation, and “Recovered” after that. We can represent the probability that an infection (perturbation) will be transmitted from one node to another by a transmission probability on the edge connecting them. When the underlying dynamical system is stochastic, this is perfectly natural; even when it is deterministic, there may be many unobserved factors that control whether a perturbation spreads from one node to another. For example, in a model for cascading failure in an electric power distribution network, the (unknown) status of protective relays can be modeled as a probability of transmitting a failure. For plant disease or pest epidemiology, such SIR-like models have previously been considered [9, 10].

Overall Transmissibility X . The weights along each edge determine the conditional probability of transmission from the source to the destination, given that the source is infectious, under a Poisson process. Specifically, for edge (i, j) from vertex i to vertex j with weight $w_{i,j} \geq 0$, the probability that i transmits to j when i is infected and j is susceptible is given by

$$p_{i,j}(y) \equiv 1 - e^{-w_{i,j}y}, \quad (1)$$

where $y \in [0, \infty)$ is a parameter that sets the scale of transmissibility. Notice that as $y \rightarrow 0$, $p_{i,j}(y) \rightarrow 0$ and as $y \rightarrow \infty$, $p_{i,j}(y) \rightarrow 1$ regardless of the value of $w_{i,j}$. Notionally, we can change variables to $\tilde{x} \equiv 1 - e^{-y} \in [0, 1]$ and think of \tilde{x} as a measure of average transmissibility. More precisely, we define $\mu(y)$ as the geometric mean probability of transmission taken across all edges:

$$\mu(y) \equiv \exp \left(|E|^{-1} \sum_{e \in E} \ln(1 - e^{-w_e y}) \right). \quad (2)$$

and take the geometric mean transmissibility as the fundamental control parameter, X , using $y = \mu^{-1}(X)$ to evaluate the probability of transmission across any particular edge.

Calibration. The distribution of outbreak sizes is the order parameter in a phase transition where the control parameter (cf. temperature) is the mean transmissibility X . We know from watching water boil¹ that the order parameter does not change uniformly across the system as the control parameter varies, but rather localized bubbles of one phase form and grow in the middle of a sea of the other phase. The resulting distributions have two or more widely separated local maxima and are poorly represented by statistics such as the mean or median. Figure 1 illustrates this behavior. In the following we characterize these multimodal distributions with a point at the third quantile and “error bars”

¹ Or, for the culinarily challenged, from equilibrium statistical mechanics.

indicating the range from median to maximum. Thus we analyze neither the worst nor the typical case, but instead the worst half of the outbreaks.

Although the transition to epidemicity depends on the network, we have chosen the values $X \in \{0.1, 0.5, 0.9\}$ for the commodity networks and $X \in \{0.05, 0.1, 0.2\}$ for the PLP networks to represent, respectively, the non-epidemic, transition, and fully epidemic phases across all the commodity networks.

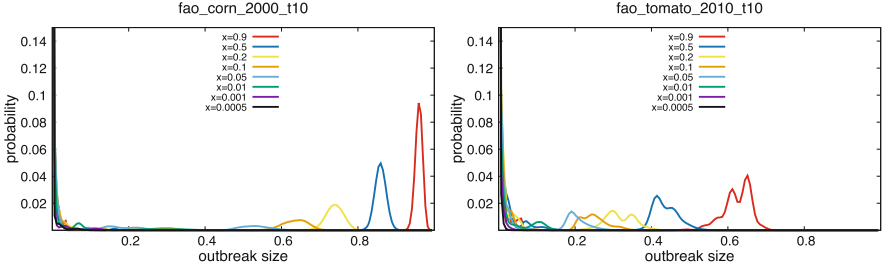


Fig. 1. Distributions of outbreak size on the 2000 corn and 2010 tomato commodity trade networks for several choices of the control parameter, the overall transmissibility X . Notice that near the transition to epidemicity, a local maximum appears at a large value of outbreak size while the local maximum at 0 shrinks but does not vanish. As X increases, the local maximum both moves to the right *and* grows larger. The tomato network shows that this local maximum may itself exhibit complicated fine structure.

2.4 Quality of Partition

We consider three aspects of subgraphs induced by a partition into communities, in addition to the conventional weighted modularity:

1. fractional cost of edges removed, i.e., the fraction of the sum of weights over all edges that is attributable to nodes in different communities.
2. inter-community influence. We define the influence of node v on node w as the probability with which an infection starting at the single node v infects w . We estimated the influence for each pair of nodes in each network using 100 *SIR* simulations per node. The inter-community influence is the fraction of the sum of over all pairs that is attributable to nodes in different communities.
3. outbreak size distribution. We estimated the size of an *SIR* outbreak using 10,000 simulations for each initially-infected node.

3 Results

Figure 2 shows the outbreak size distribution generated from 10,000 simulated outbreaks on each of 10 instances of PLP networks with varying inter-partition

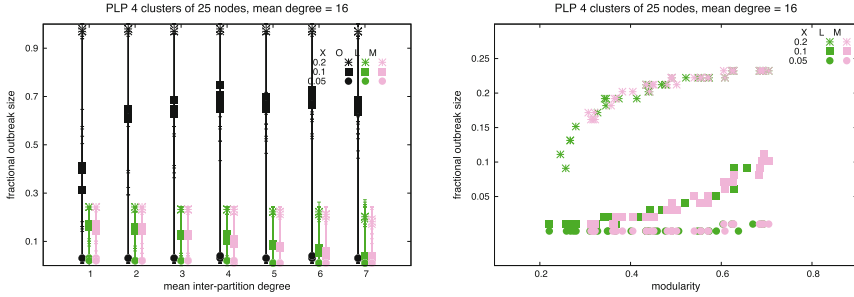


Fig. 2. Left: The range of the top two quartiles of outbreak size distribution under three partitions (‘O’ for the original planted partition, ‘L’ for Louvain, and ‘M’ for MSNR) for each of ten instances of a PLP network, as a function of the inter-partition degree k . Right: The 3rd quartile of the outbreak size distribution for the Louvain and MSNR partitions of the same PLP networks as a function of the partition’s modularity.

degree k for three values of transmissibility on the original network, MSNR cuts, and Louvain cuts. In all these cases, the Louvain algorithm recovers the planted partition; in most, but not all, of them, MSNR also recovers the planted partition. It is no surprise, then, that the outbreak size distribution and modularity (Fig. 2) are similar for each of these methods. MSNR does not maximize modularity as consistently as Louvain.

The situation is different for commodity networks. As shown in Fig. 3, MSNR’s partitions exhibit smaller outbreaks but worse modularity and influence than DL partitions on all networks. The difference in outbreak size is significant for the “corn”, “other”, and perhaps “cereal” networks but not for “tomato” networks; the difference in modularity and influence is significant for all. Furthermore, modularity and outbreak size do not seem to have a consistent or even monotonic relationship. Figure 4 shows typical differences between DL and MSNR in the same commodity network.

4 Discussion

The differences between outbreak size and modularity produced by DL and MSNR are not surprising. After all, we have allowed MSNR to construct partitions that are designed to cap the size of the largest outbreak without regard to the cost of the partition, while DL finds low-cost, high-modularity partitions regardless of outbreak size. Since both modularity and outbreak size depend on the cost of the partition, we would expect a relationship between the two. What we find surprising is that this mutual dependence on cost is not strong enough to induce a tight association between modularity and outbreak size. The strength of the association—and thus the value of using modularity as a guide to find communities whose isolation prevents large outbreaks—depends on aspects of network structure that are not yet well-understood.

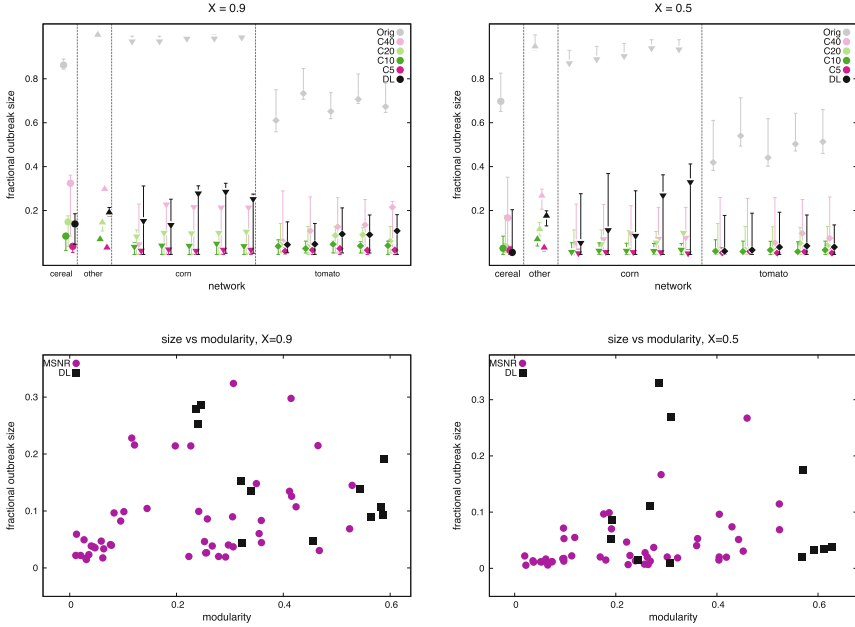


Fig. 3. Analogs of Fig. 2 for the commodity trade networks, with a range of values $C \in \{40, 20, 10, 5\}$ for the maximum allowed outbreak size parameter of the MSNR approach. Unlike for the PLP networks, there are significant differences in the outbreak size distributions for several of these networks, and the difference depends on X . Moreover, there is no obvious relation between modularity and outbreak size. Left: Overall transmissibility $X = 0.9$ Right: $X = 0.5$.

Many questions remain to be answered about this approach. Perturbations in some systems spread like a generalized epidemic in which transmission from different neighbors is not independent. In this case, spanning trees will not be the right minimal structure to define candidate communities. Resetting the state of perturbed nodes, i.e., repairing failures, may take much longer than the failure takes to spread, or repair may leave the node susceptible to failing again in the same cascade. In these cases, SI or SIS models, respectively, may be more appropriate than SIR . Simulating the distribution of outbreak sizes under these dynamics is obviously still possible, but the analysis of minimal structures may be more complicated.

Nevertheless, MSNR holds great promise for its flexibility in applications. Constraints that are often difficult to represent in other approaches can be handled without artificial manipulation of cost functions. For example, the islands in a power distribution network could be designed to include sufficient generating capacity to supply the demand of the community. Furthermore, the freedom to specify the maximum tolerable outbreak size C allows system operators to explore the efficient frontier of cost-size trade-offs.

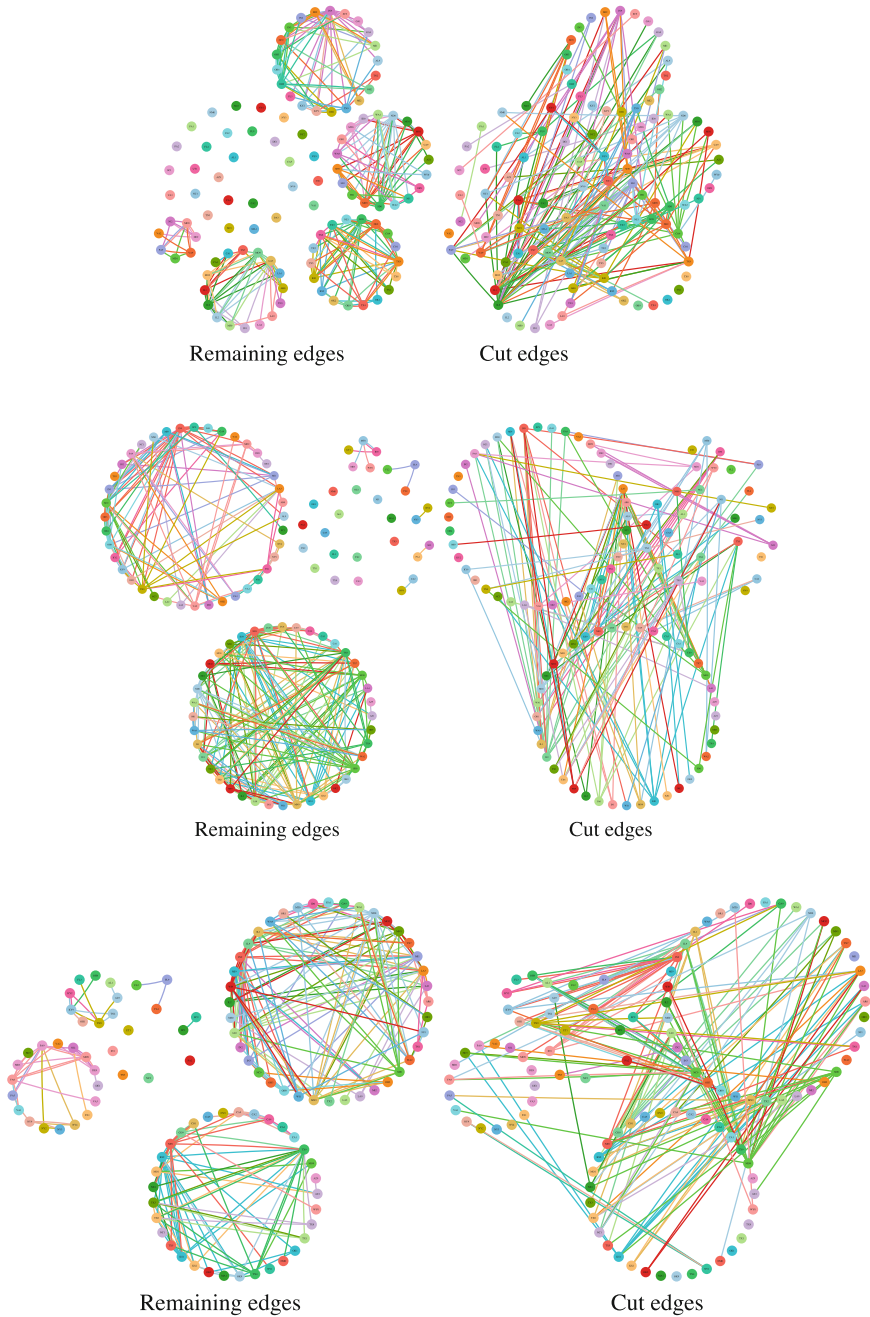


Fig. 4. Example partitions of the same network (the FAF cereal trade network) determined by Louvain (top row) and MSNR with $C = 40$ for $X = 0.9$ (middle row) and $X = 0.1$ (bottom row), with the intra-community edges on the left and inter-community edges on the right.

Acknowledgments. The author would like to acknowledge M. Nath, R. Mishra, and A. Adiga for their many helpful discussions and for constructing the commodity networks and a framework for carrying out a nontrivial experimental design and analysis. This material is based upon work supported by the National Science Foundation under Grants No. CCF-1918656 and CNS-2041952 and by grant no. 2019-67021-29933, Network Models of Food Systems and their Application to Invasive Species Spread, from the USDA National Institute of Food and Agriculture.

References

1. Berahmand, K., Bouyer, A., Vasighi, M.: Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes. *IEEE Trans. Comput. Soc. Syst.* **5**(4), 1021–1033 (2018). <https://doi.org/10.1109/TCSS.2018.2879494>
2. Domb, C.: Order-disorder statistics. ii. a two-dimensional model. *Proc. Roy. Soc. Lond. Ser. A. Math. Phys. Sci.* **199**(1057), 199–221 (1949)
3. Dugué, N., Perez, A.: Directed Louvain: maximizing modularity in directed networks. Ph.D. thesis, Université d’Orléans (2015)
4. Eubank, S., Nath, M., Ren, Y., Adiga, A.: Perturbative methods for mostly monotonic probabilistic satisfiability problems. arXiv preprint [arXiv:2206.03550](https://arxiv.org/abs/2206.03550) (2022)
5. FAF: Freight Analysis Framework (FAF) version 5 (2022). <https://faf.ornl.gov/faf5/>
6. FAO: Production and trade (2021). <http://www.fao.org/faostat/en/#data>
7. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3–5), 75–174 (2010)
8. Ghosh, R., Teng, S.H., Lerman, K., Yan, X.: The interplay between dynamics and networks: centrality, communities, and cheeger inequality. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1406–1415 (2014)
9. Gilligan, C.A.: Sustainable agriculture and plant diseases: an epidemiological perspective. *Philos. Trans. Roy. Soc. B: Biol. Sci.* **363**(1492), 741–759 (2008)
10. Gilligan, C.A., Gubbins, S., Simons, S.A.: Analysis and fitting of an SIR model with host response to infection load for a plant disease. *Philos. Trans. Roy. Soc. Lond. Ser. B: Biol. Sci.* **352**(1351), 353–364 (1997)
11. Harenberg, S., et al.: Community detection in large-scale networks: a survey and empirical evaluation. *Wiley Interdisc. Rev. Comput. Stat.* **6**(6), 426–439 (2014)
12. Leicht, E.A., Newman, M.E.: Community structure in directed networks. *Phys. Rev. Lett.* **100**(11), 118,703 (2008)
13. Malliaros, F.D., Vazirgiannis, M.: Clustering and community detection in directed networks: a survey. *Phys. Rep.* **533**(4), 95–142 (2013)
14. Mishra, R., Eubank, S., Nath, M., Amundsen, M., Adiga, A.: Community detection using Moore-Shannon network reliability: application to food networks. In: Cherifi, H., Mantegna, R.N., Rocha, L.M., Cherifi, C., Micciche, S. (eds.) *COMPLEX NETWORKS 2016 2022*. SCI, vol. 1078, pp. 271–282. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-21131-7_21
15. Moore, E., Shannon, C.: Reliable circuits using less reliable relays. *J. Franklin Inst.* **262**(3), 191–208 (1956)
16. Nath, M., et al.: Using network reliability to understand international food trade dynamics. In: Aiello, L.M., Cherifi, C., Cherifi, H., Lambiotte, R., Lió, P., Rocha, L.M. (eds.) *COMPLEX NETWORKS 2018*. SCI, vol. 812, pp. 524–535. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05411-3_43

17. Newman, M.E.: Modularity and community structure in networks. *Proc. Natl. Acad. Sci.* **103**(23), 8577–8582 (2006)
18. Palmer, W.R., Zheng, T.: Spectral clustering for directed networks. In: Benito, R.M., Cherifi, C., Cherifi, H., Moro, E., Rocha, L.M., Sales-Pardo, M. (eds.) *COMPLEX NETWORKS 2020* 2020. *SCI*, vol. 943, pp. 87–99. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-65347-7_8
19. Roth, D.: On the hardness of approximate reasoning. *Artif. Intell.* **82**(1), 273–302 (1996). <https://www.sciencedirect.com/science/article/pii/0004370294000921>
20. Valiant, L.G.: The complexity of enumeration and reliability problems. *SIAM J. Comput.* **8**(3), 410–421 (1979)
21. Wang, X., Liu, G., Li, J., Nees, J.P.: Locating structural centers: a density-based clustering method for community detection. *PLoS ONE* **12**(1), 1–23 (2017). <https://doi.org/10.1371/journal.pone.0169355>
22. Zachary, W.W.: An information flow model for conflict and fission in small groups. *J. Anthropol. Res.* **33**(4), 452–473 (1977)
23. Zhang, Y., Adhikari, B., Jan, S.T., Prakash, B.A.: Meike: influence-based communities in networks. In: *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 318–326. SIAM (2017)



Two to Five Truths in Non-negative Matrix Factorization

John M. Conroy¹, Neil Molino^{1(✉)}, Brian Baughman², Rod Gomez²,
Ryan Kaliszewski², and Nicholas A. Lines²

¹ Center for Computing Sciences, Bowie, MD 20715, USA
{conroy,npmolin}@super.org

² United States Department of Defense, Laurel, MD, USA

Abstract. In this paper we explore the role of matrix scaling on a matrix of counts when building a topic model using non-negative matrix factorization. We present a scaling inspired by the normalized Laplacian (NL) for graphs that can greatly improve the quality of a non-negative matrix factorization. The results parallel those in the spectral graph clustering work of [12], where the authors proved adjacency spectral embedding (ASE) spectral clustering was more likely to discover core-periphery partitions and Laplacian Spectral Embedding (LSE) was more likely to discover affinity partitions. In text analysis non-negative matrix factorization (NMF) is typically used on a matrix of co-occurrence “contexts” and “terms” counts. The matrix scaling inspired by LSE gives significant improvement for text topic models in a variety of datasets. We illustrate how matrix scalings in NMF can dramatically improve the quality of a topic model on three datasets where human annotation is available. Using the adjusted Rand index (ARI), a measure cluster similarity we see an increase of 50% for Twitter data and over 200% for a newsgroup dataset versus using counts, which is the analogue of ASE. For clean data, such as those from the Document Understanding Conference, NL gives over 40% improvement over ASE. We conclude with some analysis of this phenomenon and some connections of this scaling with other matrix scaling methods.

Keywords: Laplacian · Topic Modelling · NMF

Introduction

In their paper [12] the authors provide a clear and concise demonstration of the “two-truths” in spectral graph clustering. Their results prove that the choice of the first step-spectral embedding of either Laplacian spectral embedding (LSE) or adjacency spectral embedding (ASE) will identify different underlying structures, when present in a graph. The results were later made precise using Chernoff information and a stochastic block model of the underlying graphs.

The two-truth property was observed empirically while computing on connectome models of the human brain [12]. It was previously thought ASE and LSE would give comparable partitions.

The advantage of using a non-negative matrix factorization for data analysis was first noted by [11] and five years later its power was demonstrated in the Nature article [7]. In computational linguistics a matrix of co-occurrence of terms with their *context* is often employed. A context is the set of one or more terms that precede or follow a word in a collection of text [2]. Depending on the application a context could be a set of two or more consecutive words, a sentence, or one or more documents. (In this work, our main examples will be in the latter case of one or more documents as the *context* in our models.) The spectral methods applied in the text application uncover topic models, clusters of words (or more generally terms) that are related for LSE for a set of documents and ASE will tend to find “key concepts” versus “less important” information (peripheral ideas). An example using a spectral method in computation linguistics to find the key sentences was successfully used by [17].

This work tests the two-truth phenomenon on non-negative matrices, which are common in text data. Unlike the graph and their corresponding 0–1 symmetric adjacency matrices from the connectome problem, text models generally consist of integer count values and are weighted bipartite graphs as opposed to simple graphs. As the entries are counts, a non-negative matrix factorization (NMF) is often used in text analysis.

1 Bipartite Laplacian and Other Matrix Scalings

The motivation of the matrix scaling comes from a scaling of a symmetric matrix A , the adjacency matrix of a graph. Our application is a weighted bipartite graph. So, we adapt the technique with a block matrix construction. For a matrix M with all non-negative entries, representing our document-term matrix, construct the specific matrix, A , as $A = \begin{bmatrix} 0 & M \\ M^T & 0 \end{bmatrix}$, and we denote n as the number of documents and m the number of terms, giving the number of rows and columns of M , respectively.

Let $D_{r,A} = \text{diag}(A\mathbf{1})$, where $\mathbf{1}$ is the all ones vector, i.e., $D_{r,A}$ is the diagonal matrix of row sums of A . We also note that we can use this to define the column sum diagonal as $D_{c,A} = D_{r,A^T}$

The matrix A may be viewed as the adjacency matrix of a weighted bipartite graph, where we connect every word to each document by an edge weighted by the number of times that word appears in that document. With a weighted bipartite graph model, we can consider the usual variants of the graph Laplacian.

Markov: $A_{Markov} = D_{r,A}^{-1}A$

Laplacian: $L = D_{r,A} - A$

Random Walk Normalized Laplacian (RWNL): $L_{rwnl} = I - A_{Markov}$

Normalized Laplacian (NL): $L_{nl} = I - D_{r,A}^{-\frac{1}{2}}AD_{r,A}^{-\frac{1}{2}} = I - A_{nl}$

The normalized Laplacian scaling for the matrix M can be expressed as $M_{nl} = D_{r,M}^{-\frac{1}{2}}MD_{c,M}^{-\frac{1}{2}}$. In addition to this matrix scaling the algebra suggests several other representations. In particular: normalizing within the documents, i.e.,

row scaling (RS), $M_r = D_{r,M}^{-1}M$; normalizing term counts across documents, i.e. column scaling (CS), $M_c = MD_{c,M}^{-1}$; and normalizing the rows and columns independently, which is the exponential of point-wise mutual information (PWMI), $M_{rc} = D_{r,M}^{-1}MD_{c,M}^{-1}$. These four scalings in addition to the matrix M will be used as alternative matrices in our NMF factorization.

Recall that we aim to consider non-negative matrix factorization of scalings of the weighted bipartite graph. Let us posit the model [4] representation of a general matrix M

$$M = WH + E$$

where W is a m by k non-negative matrix, H is a k by m non-negative matrix, and E is a matrix of residual errors. The low-rank non-negative factorization is frequently interpreted as a topic model of k topics and the factors W and H are proportional to counts. In this way, the row normalized version of W and H are thought of as multinomial distributions expressing each document as a mixture of k topics. Each topic is a multinomial distribution over the words in the vocabulary. The mixing weights over the topics for each document are used to “assign” (associate) each document to a primary topic. The resulting factors \tilde{W} and \tilde{H} are then scaled by the inverse of the diagonal scaling applied a priori. Thus, the following pre and post scaling is done in the proposed diagonally scaled NMF.

1. *Pre-scaling:* Let D_r and D_c be the diagonal scaling matrices for the rows and columns of our data matrix M ; we then form $\tilde{M} = D_r^{-1}MD_c^{-1}$.
2. *Factorization:* Compute a non-negative matrix factors \tilde{W} and \tilde{H} where $\tilde{M} = \tilde{W}\tilde{H} + \tilde{E}$.
3. *Post-scaling:* Compute $W = D_r\tilde{W}$ and $H = \tilde{H}D_c$.

2 Computational Results

In this section we demonstrate the relative performance of the five matrix scalings when using a NMF to cluster a set of documents into topics. We first introduce the datasets, then discuss how the data are processed, and finally present the evaluation measures and results.

2.1 Three Datasets of Varying Difficulty

We illustrate the performance of the method on three datasets, which vary from relatively “well-separated” topics to “well-mixed” topics. The datasets are:

1. Document Understanding Conference 2004 (DUC 2004) multidocument summarization data. These data are 500 newswire documents which were selected to answer query needs for 50 topics. The documents were carefully chosen. First, a search engine was used to retrieve documents for each topic based on a human query. For each topic, a human selected the 10 most relevant

documents. This data set is known to be of high quality and is one of many widely used in the summarization literature.¹

2. The 20 newsgroups training data set, which consists of approximately 11K posts to one of 20 newsgroups. Each newsgroup has roughly between 400 and 600 posts. We note that the newsgroups data topics are more broadly defined than the DUC 2004 documents and will have the occasional post which may be off the group’s topic.
3. Russian Troll Twitter dataset. The English port of these data were labeled by the Clemson University researchers [9] into one of eight troll types. ‘Commercial.’ ‘Fearmonger’, ‘HashtagGamer’, ‘LeftTroll’, ‘NewsFeed’, ‘NonEnglish’, ‘RightTroll’, and ‘Unknown.’ Limiting the focus to the English tweets, there are 1648 trolls which have been assigned one of the 8 labels.² These labels define how the Twitter actors *behaved*. Each actor’s behavior would span multiple *topics* in the normal notion of a text topic model. We include these data as a challenge dataset and also as a dataset which will have text properties not generally found in newswire documents or newsgroup posts.

Here we apply the five matrix scalings: the original counts (None); column scaling (CS); row scaling (RS); pointwise mutual information (PWMI); and, normalized Laplacian (NL).

As the matrices are document-term matrices, RS scaling turns each row into the maximum likelihood multinomial distribution for a document’s vocabulary. Similarly, CS estimates the term distributions across the contexts (a.k.a. documents in this case). PWMI scales the counts by the row and column marginals, while NL scales them by the square roots of these marginals. To the extent that the spectral result of [12] carries over to NMF applied to text we would expect NL to be best at recovering latent topics.

2.2 From Data to Matrices

In this subsection we describe how the data are processed to create context-term matrices upon which we perform the five variants of non-negative matrix factorization.

Before we model the documents as a context-term matrix of counts, we must first define the notion of a *context*. The down stream task, in our case, clustering, defines how the data are separated into contexts. For the newswire dataset, a context is simply a single newspaper article. Similarly, a context for the newsgroup data is an individual post. For the Russian Troll tweets, the task is to cluster individual users, which are known *trolls*, actors who by nature of some

¹ See <https://www-nlpir.nist.gov/projects/duc/data.html> for the DUC 2004 and other DUC data. Also, see <https://tac.nist.gov/data> for similar data and their descriptions.

² <https://fivethirtyeight.com/features/why-were-sharing-3-million-russian-troll-tweets/>.

their tweets have malicious intents.³ So, here the natural notion of context is the collection of tweets from a given Twitter account.

Once the contexts are identified they are then broken into parts, which we call *terms*. Notionally, a term could be thought of as a word, but more formally, it is a function which maps documents to a vector of counts of fixed length, n , the number terms. The term definition is data dependent and is specified by a series of three parts; more formally the function can be viewed as composition of three functions. From the introduction of the vector space model [15] nearly 50 years ago it was realized that words (tokens) which occur either *too frequently* or *too rarely* do not have discriminating power and are best removed from the index. We employ information theoretic methods to remove such common and rare tokens from the index.

1. *Mapping from text to terms*: We employ two approaches to break documents into terms: first the `CountVectorizer` provided by the python module `sklearn.feature_extraction.text`⁴ and second a tokenization method popular with neural net models, `sentencepiece`. `sentencepiece` is used in Text to Text Transformer Transfer learning (T5) [13] and code supported by `huggingface.com`.⁵
2. *Removing Common Tokens*: The procedure for removing common tokens is often done with a fixed list, commonly called a *stop word list* of the most common words in the language. The approach here is algorithmic and data dependent. The choice of what tokens are removed is based on a likelihood score. The tokens are sorted by frequency and a model is posed that assumes there is a change point between the commonly occurring tokens and the tokens specific to the given data. Given a count threshold, the counts are divided into two populations. For each population the first two moments are computed and a likelihood score is computed assuming normality of the two populations. The threshold giving the maximum likelihood is then chosen to divide the populations of tokens. Tokens with the higher counts are then removed.
3. *Removing Rare Tokens*: The procedure for removing rare tokens is quite basic, but like the common token removal, data dependent. Token counts are sorted and a cumulative sum of the counts is computed and a threshold is chosen so as to keep 99% of the total of the counts.

2.3 Clustering with an NMF and Evaluating Performance

To cluster the contexts we compute a NMF for the data for a given rank, k . In the matrix factorization WH , each row of the matrix W gives non-negative

³ “In Internet slang, a troll is a person who posts inflammatory, insincere, digressive, extraneous, or off-topic messages in an online community ...” (See https://en.wikipedia.org/wiki/Internet_troll).

⁴ https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_extraction.text.

⁵ https://huggingface.co/docs/transformers/model_doc/t5.

values which are the relative strength of each “topic.” We assign each document to the topic for which it has the largest weight.⁶

As the quality of the factorization may vary based on the choice of k , we compute factorizations for a range of values of k . We use a likelihood method due to Zhu and Ghodsie (ZG) [19] to estimate the number of topics and implemented by **graspy**⁷. The singular values of the matrix M are computed and use the “second elbow,” as discovered by the likelihood method. NMF factorization using all five variants of the dimension given by this elbow as well as 10 to the left and 10 to the right of the chosen range of dimensions evaluated.

The quality levels of the resulting clusters are then evaluated using the Adjusted Rand Index (ARI) [5, 10, 14]. The Rand index allows for comparing two clustering methods, each of which may have a different numbers of clusters. The Rand index [14] is defined simply as the fraction of times the two clustering methods agree whether pairs of items are in the same cluster or not. It is based on all pairs of items. There are $\binom{m}{2}$ of these, where m is the number of the items being clustered. The adjustment part, as proposed by [5] and further studied by [10], subtracts off the expected Rand index for a random partitioning of size k , where k is the number of clusters found in the model. The values of an ARI are bounded above by 1, but can be negative when a clustering performs worse than random.

Figures 1 through Fig. 3 give the ARI for the five scalings of NMF for the range of values around the maximum likelihood estimated dimension. Here we see rather strong performance with an ARI of about 0.7 and the NL scaling gives slightly better results than the second best scaling of RS, which normalizes within the documents (Fig. 2).

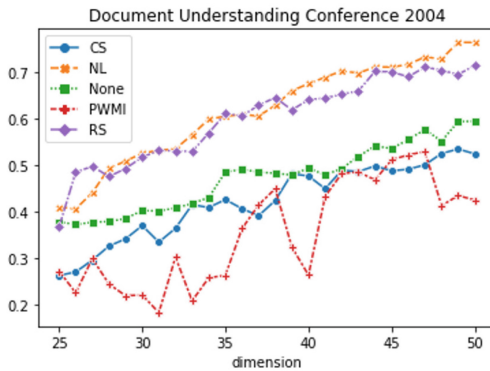


Fig. 1. Adjusted Rand Index for 5 matrix scalings: DUC 2004 Dataset.

⁶ We observe that with this simple approach the post scaling has no affect on the clustering, i.e. using \tilde{W} instead of W would yield the same clusters.

⁷ <https://github.com/bdpedigo/graspy>.

The 20 newsgroup data are overall more challenging to cluster, but here the NL scaling gives a sizable improvement over the alternative methods.

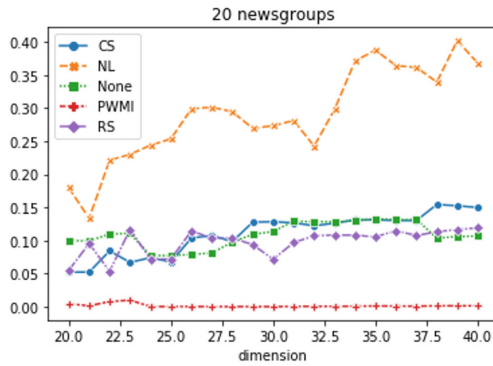


Fig. 2. Adjusted Rand Index for 5 matrix scalings:20 News Groups Dataset.

Finally, for the Twitter data, the most challenging dataset studied, the performance as measured by ARI is lower still, but with NL giving better results.

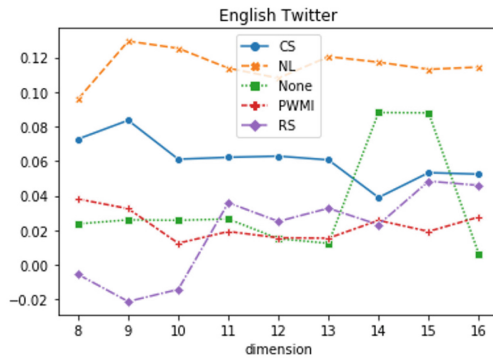


Fig. 3. Adjusted Rand Index for 5 matrix scalings: Russian Twitter Trolls.

Each of the above experiments used `CountVectorizer` to tokenize the data. To illustrate that the results can vary substantially based the tokenization we present the last dataset again comparing `CountVectorizer` with the T5's `sentencepiece`, the unigram language model. Figure 4 gives these results, where

we see both original scaling (Counts) and the normalized Laplacian (NL) improve significantly and perform comparably.⁸

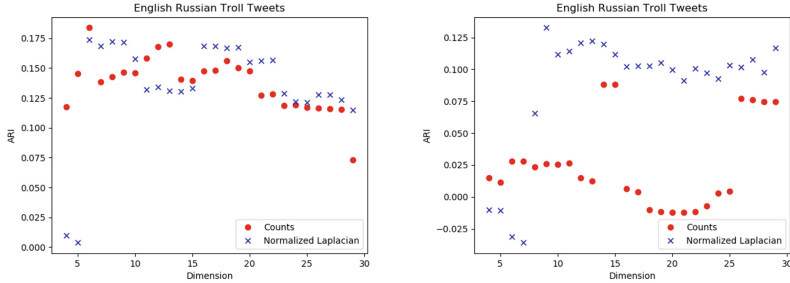


Fig. 4. Comparison of `sentencepiece` vs `CountVectorizer` Tokenization: Russian Twitter Trolls.

3 Discussion

Here we point out that the computations that we are doing are ultimately based on `sklearn`'s routine for non-negative matrix factorization, NMF. We mostly use the default parameters, but it is worth discussing a few of the major ones and the impact that they have. Specifically, we note that by default, and hence in our calculations there is no explicitly regularization. The package allows for both L_1 and L_2 or a combination of them.

The second important parameter we will spend some time investigating is that of the loss function itself. In any case, the goal is to minimize a component-wise loss function, $L : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$.

$$\min \sum_{i,j}^n L(M_{ij}, (WH)_{ij})$$

Indeed, `sklearn`'s documentation shows an input to NMF called `beta_loss`. This can be passed a float or one of three special strings. The special strings are: `frobenius`, $L_{\text{Frob}}(x, y) = (x - y)^2$; `kullback-leibler`, $L_{\text{KL}}(x, y) = x \log \frac{x}{y}$; or `itakuru-saito`, $L_{\text{IS}}(x, y) = \frac{x}{y} - \log \frac{x}{y} - 1$. They correspond to the values 2, 1, and 0 respectively.

The main point is that interaction of loss function and feature scaling is extremely important. Specifically, we note that when we scale both x and y by the same constant α the Frobenius loss is multiplied by α^2 , the Kullback-Leibler

⁸ Using the T5 tokenizer on the DUC 2004 and newsgroups data give somewhat lower performing clustering, but the relative performance of the scalings remains about the same.

loss is multiplied by a single factor of α and the Itakuru-Saito loss is unaffected. I.e.,

$$\begin{aligned} L_{\text{Frob}}(\alpha x, \alpha y) &= \alpha^2 L_{\text{Frob}}(x, y) \\ L_{\text{KL}}(\alpha x, \alpha y) &= \alpha L_{\text{KL}}(x, y) \\ L_{\text{IS}}(\alpha x, \alpha y) &= L_{\text{IS}}(x, y) \end{aligned}$$

Since we are using the Frobenius norm as our loss in this setting, we see that the loss will be dominated by large values. The quadratic scaling of the penalty means that the NMF will work very hard to match the large terms in the summed loss function.

For ease of notation, we will look at what specifically happens to the entries in these scalings for the very simple two documents with two words each setting. That is, we will be looking at 2×2 matrices. We will think of documents as the rows and the terms index the columns.

When the matrix, $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, is nothing more than the counts of words in each document, then as previously mentioned, the Frobenius norm will focus on the entries with larger values. Those will be ones that are either common words or those corresponding to longer documents. The algorithm will put much more emphasis on getting those entries correct. Obviously all of the entries here lie in \mathbb{N} .

When we consider the row scaling, $D_{r,A}^{-1}A = \begin{bmatrix} \frac{a}{a+b} & \frac{b}{a+b} \\ \frac{c}{c+d} & \frac{d}{c+d} \end{bmatrix}$, then we find that the preference for long documents is neutralized. Common words, however, are still emphasized. Also, it is clear that all of the entries in this case lie in the unit interval, $[0, 1]$.

The column scaling setting, where $AD_{c,A}^{-1} = \begin{bmatrix} \frac{a}{a+c} & \frac{b}{a+c} \\ \frac{c}{c+d} & \frac{d}{c+d} \end{bmatrix}$, is the opposite as the preference for common words is neutralized, but the preference for long documents is not. Again, the entries here lie in $[0, 1]$.

The Normalized Laplacian setting strikes a balance between the two. The preference for frequent words and for long documents are both partially neutralized. Again, the entries lie in the unit interval.

$$D_{r,A}^{-\frac{1}{2}}AD_{c,A}^{-\frac{1}{2}} = \begin{bmatrix} \frac{a}{\sqrt{(a+b)(a+c)}} & \frac{b}{\sqrt{(a+b)(b+d)}} \\ \frac{c}{\sqrt{(a+c)(c+d)}} & \frac{d}{\sqrt{(c+d)(b+d)}} \end{bmatrix}$$

The (exponentiated) point-wise mutual information scaling really came about from a very nice theoretical analysis done in [8]. The authors looked at the critical points of the objective function of word embeddings. At critical points, the embedding vectors form a low rank factorization of the pointwise mutual information matrix.

$$A_{PWMI} = \begin{bmatrix} \frac{a}{(a+b)(a+c)} & \frac{b}{(a+b)(b+d)} \\ \frac{c}{(a+c)(c+d)} & \frac{d}{(c+d)(b+d)} \end{bmatrix}$$

First of all the name comes from the fact that entries resemble the summand (or integrand) of the definition of the mutual information between two variables, $I(X; Y) = \sum_{x_i, y_j} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)p(y_j)}$.

If we look inside the log, we find the ratio of a joint probability with the product of the marginals. In this way, it can be seen as a measure of how far two random variables are from independent. In this text setting, it measures the deviation from the expected word count if we just took into account the overall word frequency and the length of the document. The calculation below normalizes the numerator and denominators in the matrix by n , the total number of words in the corpus. There, we can see that the entries are the maximum likelihood estimates of the probabilities given the corpus and match the form of the point-wise mutual information.

$$nD_{r,A}^{-1}AD_{c,A}^{-1} = n \begin{bmatrix} \frac{a}{(a+b)(a+c)} & \frac{b}{(a+b)(b+d)} \\ \frac{c}{(a+c)(c+d)} & \frac{d}{(c+d)(b+d)} \end{bmatrix} = \begin{bmatrix} \frac{\frac{a}{n}}{(\frac{a+b}{n})(\frac{a+c}{n})} & \frac{\frac{b}{n}}{(\frac{a+b}{n})(\frac{b+d}{n})} \\ \frac{\frac{c}{n}}{(\frac{a+c}{n})(\frac{c+d}{n})} & \frac{\frac{d}{n}}{(\frac{c+d}{n})(\frac{b+d}{n})} \end{bmatrix}$$

Perhaps a little less obvious is the relationship between the pointwise mutual information and the Normalized Laplacian formulation. The calculation below

$$\frac{P(X = 1, Y = 1)}{P(X = 1)P(Y = 1)} = \frac{\frac{a}{n}}{\frac{a+b}{n} \frac{a+c}{n}} = \frac{an}{(a+b)(a+c)} = \frac{\sqrt{\frac{a}{(a+b)(a+c)}}}{\sqrt{\frac{a+b}{n} \frac{a+c}{n}}} \quad (1)$$

looks at the top left entry of the matrix in particular $(A_{PWMI})_{11} = \frac{a}{\frac{a+b}{n} \frac{a+c}{n}}$ and relates it to the top left entry of the Normalized Laplacian matrix $(A_{NL})_{11} = \frac{a}{\sqrt{(a+b)(a+c)}}$. In particular, we see that

$$(A_{PWMI})_{11} = \frac{(A_{NL})_{11}}{\sqrt{\frac{a+b}{n} \frac{a+c}{n}}}$$

We further note that the denominator of the right hand side in Eq. 1 is the geometric mean of the two marginals. It has the possibility of being small and so the entries in the pointwise mutual information matrix are not necessarily constrained to lie in the unit interval and, hence, are sensitive to the quadratic scaling of L_{Frob} discussed earlier. It reintroduces a bias towards rare words or short documents.

4 Related Work

Unlike, the singular value decomposition (SVD), the solution to the NMF problem is NP-hard [18]. Not only that, but it is ill-posed; there is often not a

unique answer. For example $A = WH = WTT^{-1}H = (WT)(T^{-1}H) = \hat{W}\hat{H}$ for any invertible matrix T . This makes a similar analysis to that done in [12] less promising.

There are situations where NMF does indeed have a well-behaved unique solution. Donoho and Stodden gave a nice geometric characterization of one such case where the matrix is called separable [3]. This was further brought into the text domain in [1]. Here the separability is a relatively mild assumption. It corresponds to the presence of so-called ‘anchor’ words. An anchor word is one that is supported on one and only one topic. This is plausible for text data. For example, it is likely that the word ‘homeomorphism’ only really appears in a topic about mathematics or ‘EBITDA’ really only appears in a topic about finance. Under these assumptions, there are unique answers and efficient algorithms to find them. These algorithms are, however, sensitive to the condition number of the matrices involved. So, there is a line of work that aims to remedy these algorithms by preconditioning the matrix, M . For example, [4] uses semidefinite programming to compute a minimum volume ellipsoid. We conjecture that our Normalized Laplacian method is related to a relaxation of this problem where the ellipsoid is constrained to be axis aligned.

The authors [6] propose using diagonal row and column scaling of a matrix of counts, i.e., Sinkhorn balancing, when employing spectral clustering. The authors show that when a block structure of count matrix is present that the singular vectors of the balanced matrix exhibit a “square-wave” pattern, when reordered. They propose an algorithm using a few singular vectors to permute a matrix of counts into a block structure. A natural question is how to best extend such an approach to rectangular matrices. The authors are currently working on such an approach. Such approaches should also be studied for a non-negative factorization. The authors [16] formulate a low-rank Sinkhorn algorithm for optimal transport. They in effect are computing a NMF of a doubly stochastic matrix.

5 Conclusions

In this paper we proposed a new family of non-negative matrix factorizations inspired by normalized Laplacian and spectral graph theory. Previous work in spectral graph theory suggests that the normalized Laplacian gives rise to graph partitions which are more likely to find affinity communities in a graph versus spectral partitions based on the adjacency matrix, which tend to recover core-periphery partitions. In the non-negative matrix applications, it suggested that perhaps a normalized Laplacian scaling of the counts may give rise to better topic models. In this paper we gave strong evidence for the normalized Laplacian giving better topic models as illustrated in three text clustering datasets. In addition, we found that the choice of tokenization can significantly affect the matrix scaling and related work.

References

1. Arora, S., et al.: Learning topic models - provably and efficiently. *Commun. ACM* **61**(4), 85–93 (2018)
2. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. *J. Mach. Learn. Res.* **3**(null), 1137–1155 (2003)
3. Donoho, D., Stodden, V.: When does non-negative matrix factorization give correct decomposition into parts? In: *Advances in Neural Information Processing Systems*, vol. 16, p. 03 (2004)
4. Gillis, N., Vavasis, S.: Semidefinite programming based preconditioning for more robust near-separable nonnegative matrix factorization. *SIAM J. Optim.* **25**, 10 (2013)
5. Hubert, L., Arabie, P.: Comparing partitions. *J. Classif.* **2**(1), 193–218 (1985)
6. le Gorrec, L., Mouysset, S., Duff, I.S., Knight, P.A., Ruiz, D.: Uncovering hidden block structure for clustering. In: *Brefeld, U., Fromont, E., Hotho, A., Knobbe, A., Maathuis, M., Robardet, C. (eds.) ECML PKDD 2019. LNCS (LNAI)*, vol. 11906, pp. 140–155. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-46150-8_9
7. Lee, D.D., Seung, H.S.: Learning the parts of objects by nonnegative matrix factorization. *Nature* **401**, 788–791 (1999)
8. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: *Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems*, vol. 27. Curran Associates Inc (2014)
9. Linvill, D.L., Boatwright, B.C., Grant, W.J., Warren, P.L.: “The Russians are hacking my brain!” investigating Russia’s internet research agency twitter tactics during the 2016 United States presidential campaign. *Comput. Hum. Behav.* **99**, 292–300 (2019)
10. Nguyen, X.V., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.* **11**, 2837–2854 (2010)
11. Paatero, P., Tapper, U.: Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics* **5**(2), 111–126 (1994)
12. Priebe, C.E., et al.: On a two-truths phenomenon in spectral graph clustering. *Proc. Natl. Acad. Sci.* **116**(13), 5995–6000 (2019)
13. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**(140), 1–67 (2020)
14. Rand, W.M.: Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**(336), 846–850 (1971)
15. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. *Commun. ACM* **18**(11), 613–620 (1975)
16. Scetbon, M., Cuturi, M., Peyré, G.: Low-rank sinkhorn factorization. In: *Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9344–9354. PMLR, 18–24 July 2021

17. Steinberger, J., Jezek, K.: Using latent semantic analysis in text summarization and summary evaluation. In: Proceedings of the 7th International Conference ISIM (2004)
18. Vavasis, S.A.: On the complexity of nonnegative matrix factorization. *SIAM J. Optim.* **20**(3), 1364–1377 (2009)
19. Zhu, M., Ghodsi, A.: Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Comput. Stat. Data Anal.* **51**(2), 918–930 (2006)



Adopting Different Strategies for Improving Local Community Detection: A Comparative Study

Konstantinos Christopoulos^(✉) and Konstantinos Tsihclas

Department of Computer Engineering and Informatics, University of Patras, Patras,
Greece

kchristopou@upnet.gr, ktsichlas@ceid.upatras.gr

Abstract. In recent decades, several community detection methods have been proposed to uncover densely connected groups in a network. In many cases, there is the necessity of detecting such a group around a seed node that may be of particular topological importance or the external knowledge enable that node to be significant for the specific subgroup. In this way, local community detection algorithms can discover subgroups faster than global community detection algorithms that cover the entire network. This study aims to introduce new variants of an existing local community detection algorithm that uncover a single community and establish the positive aspects of identifying better communities. We experimentally evaluate our proposed methods in synthetic and real datasets compared to the baseline method, three state-of-the-art methods in local community detection, and a global community detection method.

Keywords: local community detection · static networks · seed

1 Introduction

Networks are widely used to represent complex systems, as they can effectively depict the natural structures and operations of many different fields, including communication, biology, and the World Wide Web. Community detection, which focuses on identifying groups of nodes that are closely connected and loosely connected to nodes in other groups in the network, has attracted the interest of many scientists. Global community detection methods aspire to discover the network partition, i.e. communities. In the last decades, many efforts have been made to identify global communities, by partitioning the whole network into subgroups. However, many researchers only focus on finding communities around specific nodes. As a consequence, the interest is focused on discovering local communities based on a few nodes [4, 18]. In addition, regarding the computational cost, local community detection (henceforth also called lcd) is the most appropriate way to identify communities in large networks e.g. social networks.

In local community detection, the typical method is to start from a set of seed nodes and detect their community. In the literature, the selection of seed

nodes can either be random [1, 8] or based on topological characteristics drawn on certain metrics [7, 16, 20] e.g. degree, closeness centrality or similarity metrics. However, these strategies are not always the best since they do not guarantee that the selected seeds are the appropriate nodes for local community detection. Authors in [2] take into account a different view of the seed in lcd, assuming that seed(s) are nodes of particular importance for their community. In this way, information from the network’s external environment (metadata) determines the nodes’ importance, which may not be related to their topological features.

In the current study, we create seven variants of an already existing static algorithm, which is presented in [19]. Consequently, we conducted experiments that used the aforementioned strategies to select seed nodes. We evaluate our proposed methods compared to the baseline method, three state-of-the-art in the field of local community detection, and a global one. Another merit of the proposed variants is that they can be extended naturally to allow for incremental updates in the local community supporting removals and insertions of nodes in this community without resorting to computation from scratch for insertions, as was the case until now.

The remaining sections are organized as follows. In Sect. 2, we review the literature on local community detection algorithms. In Sect. 3 we discuss preliminaries and give useful definitions as well as provide the baseline method and its seven variants. In Sect. 4, we present an extensive analysis of our experimental results for both synthetic and real datasets and discuss the major results of our experiments. Finally, we conclude in Sect. 5 with future work.

2 Related Work

In the literature, there are many lcd methods based on the technique to add nodes in the community on the strict condition that the inserted node maximizes the quality function. In what follows, we cite some interesting works that use the aforementioned greedy method.

The NewLCD method based on the minimal cluster is described in [22]. This algorithm has two phases, first discovers the minimal cluster and consequently finds the local community, which is extended from the minimal cluster using the LWP [15] quality function. In [6], the metric L , which is the ratio between internal and external community edges, is proposed to identify local communities. A new metric is introduced in [21]. The authors use the Closeness-Isolation (CI) metric to quantify the quality of the local community, considering the weights of the edges between neighboring nodes, which represent the similarity between two nodes. It combines the closeness of nodes within the community and the isolation of nodes from the rest of the network. An lcd method (LCDMD), which splits the lcd problem into two, is proposed in [11]. There are two key characteristics of this method: 1) The core area discovery stage, utilizing the modularity density to measure the distribution of nodes and edges, and 2) the community expansion based on the influence and the similarity between nodes and local community. Experiments in several real and synthetic datasets show high accuracy and very good results in terms of precision, recall, F1 score, and conductance.

Authors in [8], propose the local modularity R , which is the fraction of the boundary edges that are internal to the community. Here, the size of the community is predefined and the intuition is the merging of neighbor nodes to increase the R -value. The proposed approach is utilized in both synthetic and real datasets giving good results in the recall, but low precision. A new algorithm, LTE, is discussed in [14] based on a quality metric, called similarity-based tightness. Authors using the aforementioned metric, estimate the internal similarity among community nodes. It is a parameter-free method and can be used in both weighted and unweighted networks. The method achieves good results in terms of precision, recall, and NMI. The TCE algorithm that is introduced in [12], describes a method based on the expansion of the triangle community, using the fact that the edges within the community form triangles. In each step, a node is inserted in the community when the aforementioned admission is held. The method can be applied to both weighted and unweighted graphs. Furthermore, the authors suggest that the largest clique of the seed neighborhood is the best start to expand the community.

A very nice study, the Hint Enhancement Framework (HEF), is presented in [2]. HEF is a two-fold procedure. More precisely, given the seed node called hint, re-weighting and rewiring strategies are used, and an existing lcd algorithm is applied to the modified network. Here, the hint is a predefined and of particular importance node. The choice of hint encompasses external knowledge, which is derived from an expert of a certain domain or can be retrieved from the network's metadata.

Table 1. List of basic symbols used in the present work.

Symbol/Abbreviation	Description
G	The network
V	The node set of G
N	The total number of nodes in G
E	The edge set of network G
U_0	The seed node
$C(U_0)$	The initial community which contains only the seed U_0
$k_{in}^{C_i}$	The internal degree of C , when the i -th node is inserted to the community
$k_{out}^{C_i}$	The external degree of C , when the i -th node is inserted to the community C
$N(u)$	The set of neighbors of node u
LCD	The baseline Local Community Detection
$LCD - TE$	Local Community Detection using the Tree Expansion model
$LCD - TEG$	Local Community Detection using the Tree Expansion model preserving the Global condition
$LCD - E$	Local Community Detection with Expansion
$LCD - EG$	Local Community Detection with Expansion preserving the Global condition
$LCD - DEG$	Local Community Detection with Double Expansion preserving the Global condition
$LCD - DLE$	Local Community Detection with Double Linear Expansion
$LCD - Tr$	Local Community Detection with Triangles

3 Variants of a Local Community Detection with Seeds

3.1 Preliminaries and Problem Formulation

A network $G = (V, E)$ consists of a set of nodes $V = 1, \dots, n$ and a set of edges E , where $E \subseteq V \times V$ represent connections between the nodes in the network. This study considers G as an unweighted and undirected network. The neighbors of a node v are the nodes that are connected to v with a direct link, and the degree of a node is the number of its neighbors. Moreover, when two neighbors of a node v are connected, then we call this a Triangle of connections. Given a seed node A , we aim to detect the community around the seed. Seed may be a node of particular importance based on external knowledge, a node with significant topological features, or a randomly chosen node. A Local Community (LC) is defined as the community to which the seed node belongs. Hence, a network G can be divided into LC and the rest of the network $G - LC = U$.

In a greedy Local community detection method there are several quality metrics to measure the quality of a local community. In this study, the metric used is f_{monc} (henceforth called fitness score) which describes the sum of the internal community edges divided by the total sum of the internal and external edges of the community [13]. An internal edge is defined as the connection between two nodes that belong to the local community. On the other hand, an external edge is defined as the connection between two nodes, where one belongs to the community and the other belongs to its neighborhood. In Eq. 1 below, k_{in}^C and k_{out}^C are the internal and external edges of community C , respectively. In Table 1 we summarize the notation used throughout the paper.

$$f_{monc}(C) = \frac{2k_{in}^C + 1}{2k_{in}^C + k_{out}^C}, \quad (1)$$

Algorithm 1. Baseline *LCD*

Input: G, U_0

```

 $C(U_0)$ 
 $f_C = 0$ 
while add new nodes in  $C$  do
   $max = 0$ 
   $f_{max} = 0$ 
  for  $u \in N(C)$  do
     $f_u \leftarrow f_{(C \cup \{u\})}$ 
    if  $f_u > f_C$  and  $f_u >$ 
 $f_{max}$  then
       $f_{max} \leftarrow f_u$ 
       $max \leftarrow u$ 
    end if
  end for
   $C \leftarrow C \cup \{max\}$ 
   $f_C \leftarrow f_{max}$ 
end while

```

In what follows, the baseline LCD framework is presented in more detail, see Algorithm 1. Given a network G and a seed U_0 , each time we aim to add a new member node in C . Initially, the community contains only the seed, $C(U_0)$, and its fitness score is $f_C = \frac{1}{k_{out}^C}$. Consequently, the algorithm searches, between all community neighbors, to find the node whose potential insertion in the community maximizes its fitness score. When the algorithm finds the node whose affiliation offers the maximum fitness score for the community, the next step is to check whether the estimated fitness score is higher than the fitness score of the current community state i.e. without the new node. If this condition is satisfied, then the new node is added to the community. In this way, whenever a new node is

added in C the fitness score of C increases, and at the end of the process, we get the fitness scores in increasing order. For instance, let's suppose that a community C contains only the seed U_0 . Searching the neighbors of U_0 , we find that U_1 offers the maximum fitness score for C , compared to all neighbors of U_0 , and the insertion of that node in C increases the community fitness score, e.g. $f_{U_1} > f_{U_0} = f_C$. Subsequently, the node is added to C and the new community fitness score is $f_C = f_{U_1}$. The process continues until no new candidate node can increase the fitness score of C . In Table 2 we summarise the process.

Table 2. Community evolution depending on the fitness scores order.

Sequence	0	1	2	n
Nodes	U_0	U_1	U_2	U_n
Internal edges	$K_{0,in}$	$K_{1,in}$	$K_{2,in}$	$K_{n,in}$
External edges	$K_{0,out}$	$K_{1,out}$	$K_{2,out}$	$K_{n,out}$
Fitness score of C	f_{U_0}	f_{U_1}	f_{U_2}	f_{U_n}

3.2 Proposed Variants

Given an unweighted network G , with a set of nodes V and a set of edges E , we propose seven methods to detect local communities. All methods are variants of the baseline LCD. In what follows, we give an extensive description of each one.

1. *LCD-TE*: The Local Community Detection method using the Tree Expansion model (LCD-TE) is the first variant of LCD. The basic idea of this method is that we can add in C at most two nodes each time, and the community is represented as a tree. For example, given a seed node U_0 and its neighbors U_1, U_2, U_3, U_4 and U_5 , we apply the Algorithm 1, see Fig. 1. Let's assume that after the first iteration the fitness scores, f_{U_i} , of the possible community members are the following: $f_{U_3} > f_{U_1} > f_{U_4} > f_{U_2} > f_{U_5} > f_{U_0} = f_C$. From the above inequality, we see that the community reaches the maximum fitness score if U_3 is added in C and the second maximum if U_1 is added in C . The

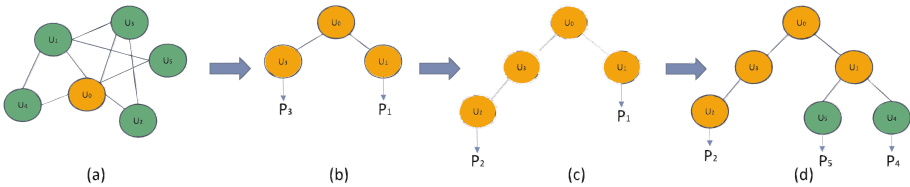


Fig. 1. Illustration of LCD - TE(G) process. a) The initial network G , b) Community of 3 nodes, c) Community of 4 nodes, and d) Community of 6 nodes

next step is to insert them in C and represent its members as a tree that consists of the parent U_0 and the two children U_3 and U_1 , where U_3 is the left and U_1 the right child. We keep this order because $f_{U_3} > f_{U_1}$ and so, the node that offers the maximum fitness score is added always as the left child.

Taking into account the formed community in Fig. 1b, which consists of two community groups, (U_0, U_1) and (U_0, U_3) , the next step is to apply the Algorithm 1 to these groups individually. We always start from the left-hand side group because the left children always offer the maximum fitness score, and we do this as a reward for these nodes. One restriction is that the neighborhood of each group consists only of neighboring nodes that are not members of other community groups.

Applying the Algorithm 1 to the group (U_0, U_3) , holds that $f_{U_2} > f_{U_3} > f_{U_4}$. Then, only node U_2 is added to this group as a leaf, see Fig. 1c. Now the formed groups are $P_2 = (U_0, U_3, U_2)$ and $P_1 = (U_0, U_1)$, where P_i is the group of nodes from root U_0 to leaf U_i . Consequently, Algorithm 1 is applied to P_2 . If no other node can be added to this group, we apply the Algorithm 1 to the next group, P_1 . Let's assume that $f_{U_5} > f_{U_4} > f_{U_1}$, then both nodes U_5 and U_4 are added as leaves under the parent U_1 . The new formed groups now are $P_5 = (U_0, U_1, U_5)$ and $P_4 = (U_0, U_1, U_4)$, see Fig. 1d. We observe that any time we work in a group, two community groups may be formed e.g. from P_1 , the groups P_5 and P_4 are formed. Afterward, Algorithm 1 is applied to P_5 . Due to the fact that no other node can be added to group P_5 , we apply Algorithm 1 to group P_4 . Assuming again that no other node can be added to P_5 , the process is terminated since there is no other community group. Thus, we always apply Algorithm 1 to the left-hand side group, and the process terminates when no other node can be inserted into any community group. The termination condition is the same for all methods.

2. *LCD-TEG*: The Local Community Detection method using the Tree Expansion model preserving the Global condition (LCD-TEG), is the second variant of LCD. The process of this variant is the same as the LCD-TE. The main difference is one extra condition that holds whenever we check whether a node should be added to a group P_i . For this reason, we keep a record of the global fitness score. More precisely, we calculate the fitness score of the entire community f_C , merging the member nodes of all groups. Let's make the assumption that nodes U_i and U_j are the two candidate nodes for insertion, and hold that $f_{U_j} > f_{U_i}$. First, we check for node U_i whether the extra condition $f_{U_i} > w \cdot f_C$ is satisfied (w is a threshold that will be defined in the experiment section). If the extra condition holds, node U_i is added as a leaf under the corresponding group and the entire community fitness score f_C is recalculated. Consequently, we do the same for node U_j . If for both nodes U_i and U_j the global condition is satisfied, U_j is added as the left child and U_i as the right child. We always check last the node that offers the maximum fitness score to the current community group.
3. *LCD-E*: The Local Community Detection Expansion Method (LCD-E) is the third variant of LCD. This method is more simplified since there is no tree

expansion; instead, we add as leaves under the seed the first two neighbors whose affiliation in $C(U_0)$ offers the maximum fitness score. Then, for the formed two groups (see Fig. 1b) we use Algorithm 1 for each one individually, and each time only one node can be added to the corresponding group. Like the previous methods, we start always from the left-hand side group.

4. *LCD-EG*: The Local Community Detection Expansion method preserving the Global condition (LCD-EG), is the fourth variant of LCD. The LCD-EG method is the same as LCD-E. The only difference is the global condition. Thus, two community groups are formed but the insertion of a new node U_i in the corresponding group depends also on the fitness score of the entire community f_C , e.g. if $f_{U_i} > w \cdot f_C$ is satisfied.
5. *LCD-DEG*: The Local Community Detection with Double Expansion method preserving the Global condition (LCD-DEG), is the fifth variant of LCD. In this method, two community groups are formed like in LCD - EG and LCD - E. On the other hand, at most two nodes can be added to each community group, preserving the global condition $f_{U_i} > w \cdot f_C$. The key difference of this method compared to LCD - TEG is that when two nodes should be added to a group, they are added one after the other, and not as leaves in a tree. In this way, no more new groups are formed except for the initial two.
6. *LCD-DLE*: The Local Community Detection with Double Linear Expansion Method (LCD-DLE) is the sixth variant of LCD. This is the most simplified variant of all seven. The process is the same as the baseline LCD. The only difference is that, unlike LCD, each time we add at most two nodes to the C . Thus, the two neighbors that offer the maximum fitness score to C and their fitness scores are greater than the score of the current community, are inserted into C . When two nodes should be inserted in C , we always add last the node that offers the maximum fitness score to C .
7. *LCD-Tr*: The Local Community Detection with Triangles (LCD-Tr) method is the seventh and last variant of LCD. In this approach, given a seed U_0 , we find its neighboring nodes that form triangles with U_0 . After this preliminary work, all these nodes are added to the C . Consequently, the fitness score of the developed community is estimated and the baseline LCD is applied.

Time Complexity

Given that n is the community size and d is the average degree, the time complexity of the LCD [19] is given $O(n^2d)$. For both LCD-TE and LCD-TEG, the execution time is equal to $O(mn^2d)$, where m is the number of different community groups in which the Algorithm 1 is applied. Indeed, $m \ll n$ since the number of leaves in the tree is much less than the number of nodes in C . The above analysis is very pessimistic, while the size of each community group is less than the size n of the entire community C . Subsequently, the time complexity of both algorithms is $\approx O(n^2d)$.

Regarding the LCD-E, LCD-EG, and LCD - DEG methods, the complexity is nearly the same as LCD. In these methods, only two community groups are created. Thus, the time that we need to run the algorithm is $O(2n^2d)$, so it would

still be $O(n^2d)$. For the LCD-DLE method in the worst case, the complexity is equal to the LCD, since in this method there is only one community group. This method can add two nodes in C , and consequently, in the best case the time can be reduced to half, compared to the LCD. Lastly, the LCD-Tr method needs extra time to find the nodes which form triangles with the seed. For this preliminary work, the time complexity is equal to the degree of the seed. Given the average degree d , the overall time will be $O(d + n^2d)$. Consequently, in the worst case it would still be $O(n^2d)$.

4 Experiments

4.1 Experiment Design

To evaluate the results of our proposed variants, we use precision, recall and the F1 score as evaluation metrics. Given a detected community and the corresponding ground truth, precision is equal to the ratio between correctly detected nodes and the total number of detected nodes. The recall is the fraction of relevant nodes that are detected successfully. F1 score or measure F is the harmonic mean of precision and recall [10]. The harmonic mean is used instead of the simple average because it combines precision and recall in a way that gives equal weight to both.

We compare the proposed variants with the baseline LCD and three of the most effective algorithms for detecting local communities, LTE [14], NewLCD [22], and TCE [12]. Additionally, we use the Louvain method [5] on the real datasets. This method attempts to find the partition that maximizes the modularity of the network, using a greedy heuristic that iteratively moves nodes to different communities and merges communities into larger ones. We chose Louvain because it is fast, scalable, and has been used in various domains such as social network analysis, biology, and physics.

To experimentally assess the fitness score threshold w , for the methods that preserve the global condition (LCD - TEG, LCD - EG, and LCD - DEG), we use $w = 0.5$, $w = 0.7$, and $w = 0.9$. Experiments for $w = 1$ are also conducted, but not presented in this study since the results are not worth mentioning. We conducted our experiments on an Intel Core i7 processor with a speed of 2.9 GHz and 16 GB of RAM. We used Python programming language and the NetworkX, igraph, and numpy libraries to implement the methods.

4.2 Experiments on Synthetic and Real Datasets

In this study, we use both synthetic and real datasets. Synthetic datasets that have been used are created using the RDyn [17] generator. RDyn generator creates dynamic networks concerning real network attributes with multiple time-dependent ground-truth partitions. Regarding the parameters, the generator uses two user-defined parameters. The first is the number of nodes and the second is the number of iterations. Each iteration consists of edge insertions and

deletions, and after each iteration, the corresponding ground-truth communities are generated. Considering we work in static networks, the whole synthetic network is loaded and we assume as ground-truth communities the last created communities. In our experiments, we use two different datasets produced by the RDyn generator. The basic attributes of the synthetic datasets are described in Table 3.

Regarding the real datasets, we tested our proposed variants on three quite large networks [9]. More specifically, the first dataset we used is the email Eu core, which was generated using email data from a large European company. Edges represent email communications between employees (nodes). The ground-truth communities are provided and represent the departments to which each employee belongs.

The second real dataset is the Amazon product co-purchasing network. The network is created by spidering the Amazon website. The network is based on the Consumers Who Bought This Item Also Bought feature of the Amazon website. An undirected edge between two products i and j means that both products are regularly co-purchased. The provided ground-truth community represents a product class.

The third and last dataset is the DBLP collaboration network. The DBLP dataset contains thorough records of computer science papers. The generated undirected co-authorship network represents the connection between two authors (nodes) i and j as the co-publication of at least one paper. Regarding the ground-truth communities, the authors who published in a specific journal or conference shape a community. The basic characteristics of these datasets are described in Table 4.

Table 3. Properties of the synthetic datasets. **Table 4.** Properties of the real datasets.

Dataset	Nodes	Iterations	Edges
<i>SD1</i>	1000	5000	4917
<i>SD2</i>	2000	2000	16413

Dataset	Nodes	Edges
email-Eu	1005	25571
Amazon	334863	925872
DBLP	317080	1049866

To begin with, for both synthetic datasets, experiments were carried out using several nodes of low, average and high-degree centrality, and the most noteworthy results are presented. Regarding the figures, the y-axis represents the F1 score and the x-axis the different methods. As discussed in the previous subsection, we compared our LCD variants with four baseline lcd algorithms, LCD, LTE, TCE and NewLCD. The obtained F1 score for each of the proposed methods represents the average score of the several nodes, which have been used as seeds. The dashed line in each of the figures serves as a separator between the proposed and the baseline methods.

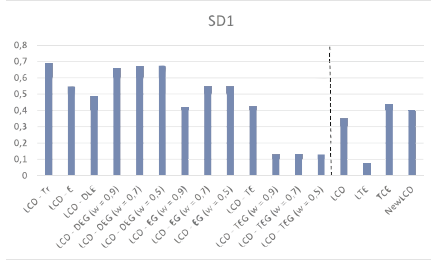


Fig. 2. The average F1 score of the proposed methods compared to the baselines.

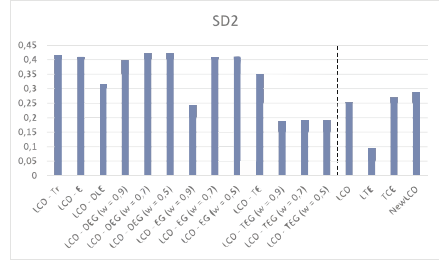


Fig. 3. The average F1 score of the proposed methods compared to the baselines.

In Fig. 2 we present the results of the experiments with SD1 dataset. This network consists of $n = 1000$ nodes and $m = 4917$ edges. Analyzing the results we realize that most of the proposed methods outperform the baselines. More precisely, we obtain an improvement of 25% for the F1 score when the LCD - Tr method is used. On the other hand, the LCD - TEG method presents the worst results since recall performance is too low. Regarding the approaches that preserve the global condition (LCD - DEG, LCD - EG and LCD - TEG), we see that for $w = 0.7$ and $w = 0.5$ we get the best results.

In Fig. 3, we present the results of the experiments with SD2 dataset. This network is quite dense since it consists of $n = 2000$ nodes and $m = 16413$ edges. The results of this experiment are similar since the maximum improvement for the F1 score between LCD - DEG and NewLCD, is 13%. All the proposed methods perform very well compared to the baselines. The LCD - TEG method obtains again poorly results in the F1 score but is much better compared to the SD1 dataset. The LCD -TEG method can discover small partitions since the detected communities consist only of a few nodes. Moreover, regarding the methods that preserve the global condition, when $w = 0.7$ we obtain again the best results. On the other hand, on both synthetic datasets, when $w = 0.9$ the F1 score is decreased since we observe a significant reduction in the recall metric. So, after evaluating the parameter w , for the rest of the experiments, we use $w = 0.7$ as the fitness score threshold.

In Figs. 4, 5, 6, and 7, we present the results of the experiments of LCD variants utilizing the real datasets. We use the same representation as in synthetic datasets. Besides the four baseline algorithms, for the sake of comparison, we also use the global community detection algorithm Louvain. Louvain performs very well only on the email-EU-core network, since in the rest networks the results are below average.

The results of the experiments with the email-EU-core network are presented in Fig. 4. For this network, we use as seed a node of particular importance, retrieved from metadata [3]. The selected seed is the manager of a certain department, and applying our proposed variants to the chosen seed, we discover the corresponding department. For this network, Louvain performs better than the

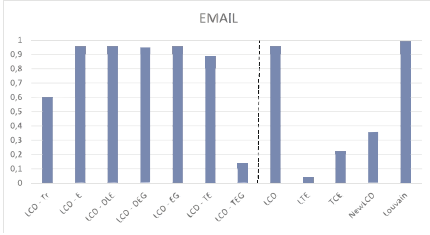


Fig. 4. The F1 score of the proposed methods compared to the baselines.

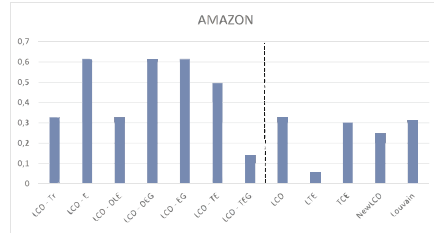


Fig. 5. The F1 score of the proposed methods compared to the baselines.

rest of the algorithms. More precisely, its F1 score is almost 98% and LCD, LCD - E, LCD - DLE, LCD - DEG and LCD - EG follow with nearly 96%. On the other hand, the LCD - Tr method has low precision and, for this reason, its F1 score is reduced. In addition, as a result of the extremely low recall, the LCD - TEG method performs below average. In this experiment, we see that the obtained results of the proposed variants are almost equal to LCD. However, execution time is refined in all of the proposed methods. More precisely, the F1 score of LCD - TE is 88% but its execution time is 3 times faster than LCD. Similar examples can also be given for the rest of the proposed algorithms. On the other hand, there is a slight decrease in time efficiency for some of the proposed approaches, compared to LTE, TCE and NewLCD. Lastly, with regard to the F1 results, LTE, TCE and NewLCD algorithms do not perform well, and their scores are below the average.

In Fig. 5 we present the results of the experiments with the Amazon network. For this network, we work in a different way to choose the seed node. Initially, we randomly choose a quite large community and find the central node of the chosen community, using the degree centrality as a measure. For this network, three of the proposed methods (LCD - E, LCD - DEG and LCD - EG) reach the score of 62%, which is the maximum F1 score for this experiment. The result, on average, is improved by 28% for recall, 13% for precision and 30% for F1 score. One more time we observe the extremely low recall value of the LCD - TEG method, and as a consequence, its F1 score is decreased more than the rest proposed variants. In this experiment, both the retrieved results and execution time realise considerable improvement, compared to the baseline methods.

The last two experiments are conducted for the DBLP network, and in Figs. 6 and 7 we present the obtained results. For the experiment in Fig. 6, we randomly choose a community and find its central node using the degree centrality as a measure. Utilizing that node as a seed, five of the proposed methods have remarkable results. More precisely, the F1 score for the LCD - Tr, LCD - E, LCD - DEG, LCD - EG and LCD - TE methods is almost 75%. The result, on average, is improved by 50% for the F1 score and more than 40% for recall, compared to the baselines. On the other hand, the low F1 score of the LCD - DLE and LCD - TEG methods reflects to the 10% of their recall value.

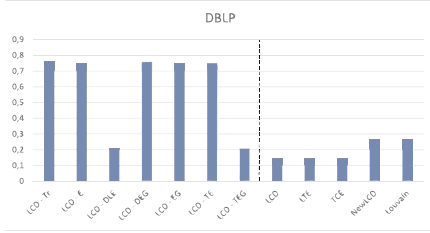


Fig. 6. The F1 score of the proposed methods compared to the baselines.

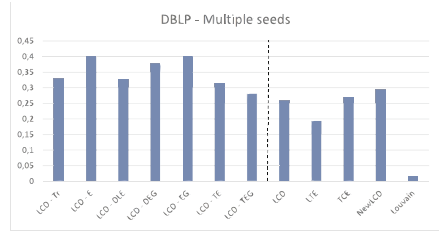


Fig. 7. The average F1 score of the proposed methods compared to the baselines.

Finally, in Fig. 7, we present the results of the experiments with the DBLP network, using multiple nodes as seeds. More precisely, we choose randomly a community and use all its nodes, one by one, as seeds. Consequently, Fig. 7 shows the average F1 score for each of the proposed methods. Once again, the results show the dominance of our proposed methods, since almost all approaches outperform the baseline methods. In more detail, the result is improved by 11% for the F1 score and by 23% for recall, compared to the baselines. A notable result of this experiment is the quite high F1 score of LCD - TEG, compared to the previous experiments. The explanation for this is that the chosen community is not large, and the LCD -TEG method can discover small partitions since its detected communities consist only of a few nodes. One more time, both the retrieved results and execution time realize considerable improvement, compared to the baseline methods for both experiments in the DBLP network.

Discussion

The experiments demonstrate that the proposed LCD variants improve the community detection results. Experiments are conducted in both synthetic and real datasets using different seed selection methods each time. Furthermore, an experimental evaluation is done for the fitness score threshold w , selecting the threshold that gives the best results for our proposed models. The current study is focused on differences in F1 score between the proposed variants and the baseline methods. The results show that the proposed variants outperform the baselines in terms of the F1 score.

On average, all the proposed methods perform better than the baselines, except for LCD-TEG. Examining carefully the retrieved results, we observe that the LCD-TEG method experiences better results for the recall metric when the detected community is small. This indicates that the LCD-TEG method is suitable for discovering communities consisting only of a few member nodes. In addition, the execution time of the proposed approaches, in the worst case, is at least equal to the LTE, TCE, and NewLCD methods when the network is sparse, e.g. Amazon and DBLP. On the other hand, when the network is dense, e.g. SD2 and email network, there is a slight decrease in time efficiency for some of the proposed approaches. Finally, Louvain is the most time-consuming algorithm.

5 Conclusions and Future Scope

In recent years, scientists have been intrigued by the concept of local community detection. This paper focuses on the identification of local communities that contain significant nodes, referred to as seeds. The importance of a node is obtained from metadata or extracted from the topological characteristics of the network. To accomplish the process of local community detection, we propose seven variants of an existing lcd algorithm. An experimental evaluation of the proposed variants is conducted, using two synthetic and three real datasets. For the sake of comparison, we used four local and one global community detection algorithm, as the baseline methods. Our findings demonstrate that the proposed variants outperform the baseline methods in terms of the F1 score. In addition, in most cases, we obtain better execution time.

This work contains results on static networks and we intend to extend these results along the following axis: 1. Expand the proposed variants to discover local communities on dynamic networks. 2. The proposed strategies can be extended to support removals and insertions of nodes in the local community without resorting to computation from scratch for insertions, as was the case until now. 3. In addition to detecting local communities, we intend to identify anomalies in dynamic networks.

Acknowledgment. “This research was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the “2nd Call for H.F.R.I. Research Projects to support Faculty Members & Researchers” (Project Number: 3480).”

References

1. Bagrow, J.P., Boltt, E.M.: Local method for detecting communities. *Phys. Rev. E* **72**(4), 046,108 (2005)
2. Baltso, G., Tsihlias, K., Vakali, A.: Local community detection with hints. *Appl. Intell.*, 1–22 (2022)
3. Bharali, A.: An analysis of email-eu-core network. *Int. J. Sci. Res. Math. Stat. Sci.* **5**, 100–104 (2018). <https://doi.org/10.26438/ijstrmss/v5i4.100104>
4. Bian, Y., Luo, D., Yan, Y., Cheng, W., Wang, W., Zhang, X.: Memory-based random walk for multi-query local community detection. *Knowl. Inf. Syst.* **62**, 2067–2101 (2020)
5. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**(10), P10,008 (2008)
6. Chen, J., Zaïane, O., Goebel, R.: Local community identification in social networks. In: 2009 International Conference on Advances in Social Network Analysis and Mining, pp. 237–242. IEEE (2009)
7. Chen, Q., Fang, M.: Community detection based on local central vertices of complex networks. In: 2011 International Conference on Machine Learning and Cybernetics, vol. 2, pp. 920–925. IEEE (2011)
8. Clauset, A.: Finding local community structure in networks. *Phys. Rev. E* **72**(2), 026,132 (2005)

9. Collection, S.L.N.D.: email-eu-core network (2007). <https://snap.stanford.edu/data/email-Eu-core.html>
10. F1 score lemma: F1 score lemma – Wikipedia, the free encyclopedia (2020). https://en.wikipedia.org/wiki/F1_score
11. Guo, K., Huang, X., Wu, L., Chen, Y.: Local community detection algorithm based on local modularity density. *Appl. Intell.* **52**(2), 1238–1253 (2022)
12. Hamann, M., Röhrs, E., Wagner, D.: Local community detection based on small cliques. *Algorithms* **10**(3), 90 (2017)
13. Havemann, F., Heinz, M., Struck, A., Gläser, J.: Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels. *J. Stat. Mech. Theory Exp.* **2011**(01), P01,023 (2011)
14. Huang, J., Sun, H., Liu, Y., Song, Q., Wening, T.: Towards online multiresolution community detection in large-scale networks. *PloS One* **6**(8), e23,829 (2011)
15. Luo, F., Wang, J.Z., Promislow, E.: Exploring local community structures in large networks. *Web Intell. Agent Syst. Int. J.* **6**(4), 387–400 (2008)
16. Moradi, F., Olovsson, T., Tsigas, P.: A local seed selection algorithm for overlapping community detection. In: 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), pp. 1–8. IEEE (2014)
17. Rossetti, G.: Rdyn: graph benchmark handling community dynamics. *J. Complex Networks* **5**(6), 893–912 (2017)
18. Veldt, N., Klymko, C., Gleich, D.F.: Flow-based local graph clustering with better seed set inclusion. In: Proceedings of the 2019 SIAM International Conference on Data Mining, pp. 378–386. SIAM (2019)
19. Zakrzewska, A., Bader, D.A.: A dynamic algorithm for local community detection in graphs. In: Proceedings of the 2015 IEEE/ACM/International Conference on Advances in Social Networks Analysis and Mining 2015, pp. 559–564 (2015)
20. Zhang, T., Wu, B.: A method for local community detection by finding core nodes. In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 1171–1176. IEEE (2012)
21. Zhao, W., Zhang, F., Liu, J.: Local community detection via edge weighting. In: Information Retrieval Technology: 12th Asia Information Retrieval Societies Conference, AIRS 2016, Beijing, China, November 30–December 2, 2016, Proceedings 12, pp. 68–80. Springer (2016)
22. Zhou, Y., Sun, G., Xing, Y., Zhou, R., Wang, Z., et al.: Local community detection algorithm based on minimal cluster. *Applied Computational Intelligence and Soft Computing* 2016 (2016)



Pyramid as a Core Structure in Social Networks

Wenruo Lyu^(✉) and Liang Zhao

Graduate School of Advanced Integrated Studies in Human Survivability, Kyoto University, Kyoto 606-8306, Japan

lyu.wenruo.y36@kyoto-u.jp, liangzhao@acm.org

Abstract. The significance of pyramid structures in dominating social networks is often mentioned, but the concept lacks a precise definition and comprehensive empirical analysis. To fill this gap, this paper offers a rigorous definition of pyramid structure, characterizing it as an expanding rooted tree. Empirical evidence, drawn from an extensive study of real-world social networks, corroborates existing hypotheses about the prevalence of large pyramid structures in such networks. Additionally, we identify a unique network characteristic: pyramid sizes vary across different types of networks. This finding suggests that existing models may not be sufficient for accurately representing real-world social networks and calls for the development of a new model. The paper also explores more characteristics and applications of pyramid in developing network metrics and structures, touching on aspects like degree distribution, clustering coefficient, core-periphery structures, and small-world networks.

Keywords: social network · pyramid structure · representative nodes · small world · core-periphery structure

1 Introduction

Since the late 1990s, network analysis has emerged and rapidly attracted the interest of an increasing number of researchers from diverse disciplines [1, 5, 8, 20, 30]. A variety of tools and theories have been developed for that purpose, including but not limited to degree distribution, clustering coefficient, average distance, and core-periphery structure, as well as the Watts-Strogatz (WS) model [29] and the Barabási-Albert (BA) model [3] (see [2] for a survey).

In the existing literature, the term “hierarchical structure” is frequently employed. Typically, this refers to the categorization of network nodes based on specific structural properties, such as degree, reachability within k hops, or participation in a given number of triangles. Nodes sharing the same structural characteristic are grouped into the same partition [6, 14, 16].

However, the concept of “pyramid structure” is often less precisely defined, despite its recurring appearance in academic studies. Sometimes it is ambiguously used or treated as a synonym for hierarchical structures (e.g., [6, 12]). It’s crucial to distinguish between the two: a pyramid structure expands from a single point at the top and grows progressively larger toward the bottom, whereas

hierarchies can manifest in various shapes. For instance, the Great Pyramid of Giza serves as an iconic example of a pyramid, not a “Great Hierarchy of Giza.”

The distinction between pyramid and hierarchical structures is not merely semantic; it has practical implications for the study of networks. Pyramid structures, with their characteristic expanding or growing features, offer specific advantages for network analysis. For example, Taagepera [24] employed a pyramid-based model (Fig. 1(a)) to explore “social mobilization,” leveraging its structural properties to formulate a mathematical theory that enabled him to estimate the size of an assembly based on the total population and adult literacy rates.

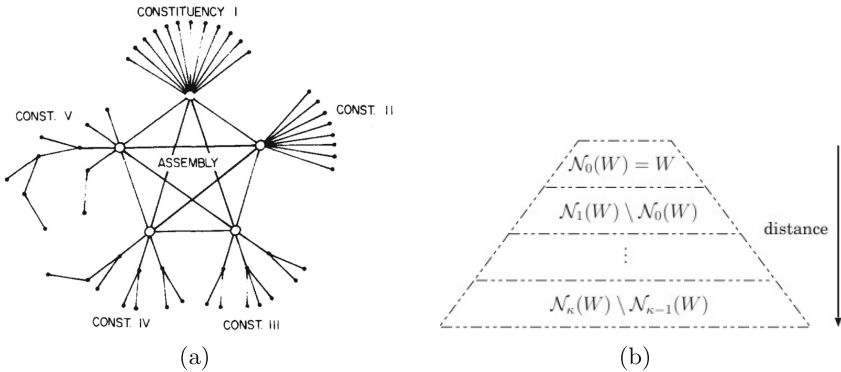


Fig. 1. Historical illustrations of pyramid structures in representing in a social network. Source: (a) Taagepera [24] (Fig. 2); (b) Zhao and Peng [30] (Fig. 2).

Analogously, Zhao and Peng [30] studied the size of representative nodes in social networks with a hypothesis on the existence of *large* pyramid structures. They provided a more rigid definition as follows (see Fig. 1(b) for an illustration). Rooted at a (representative) node set $W \neq \emptyset$, a pyramid consists of $\kappa \geq 1$ layers of nodes. The k -th layer $\Gamma(W; k - 1)$ consists of all nodes of distance $k - 1$ from W , such that $|\Gamma(W; k)| \approx c|\Gamma(W; k - 1)|$ for some constant $c > 1$, $1 \leq k \leq \kappa$. They call it a pyramid of an expansion rate c . With some reasonable assumptions, they developed the world-first theoretical analysis to explain the phenomenon that the size of a national parliament scales to the 0.4-th power of the population [30].

However, while Taggepera’s work [24] offered a theoretical model, it lacked empirical validation. Similarly, Zhao and Peng [30] only studied one real-world social network, and presented an overly simplistic model that do not fully capture the complexities of actual situations.

To remedy this shortfall, our study introduces a novel and practical definition of pyramid structure. Empirical evidence from a broad range of real-world social networks reinforces the hypotheses previously suggested. Importantly, we uncover a new structural characteristic: pyramid sizes show distinct patterns across different types of networks. They tend to be small in road networks, large in networks generated by the WS or BA models, and exhibit a diverse range in

social networks. This pattern points to the need for a new, more nuanced model to accurately represent real-world social networks. Moreover, we demonstrate that networks can also be characterized by the average minimum and maximum expansion ratios of pyramids, the proportion of large basic pyramids with a minimum expansion ratio that is no less than 2, and the identification of the pyramid-based core.

Furthermore, our findings have implications for the design of network metrics and structures. Elements like degree distribution, clustering coefficient, core-periphery structure, and the concept of small-world networks can all be reconsidered in the light of our more refined understanding of pyramid structures. Consequently, our work not only fills a gap in the current academic discourse but also provides valuable insights for the development of future network analysis tools and methodologies.

2 Preliminary

Let $G = (V, E)$ denote an undirected graph with a set V of n nodes and a set E of m edges. Without loss of generality, we assume that G is simple and connected. The degree of a node v is denoted by $\deg(v) = |\{(u, v) \in E\}|$. The distance between nodes u and v in graph G is denoted by $\text{dist}_G(u, v)$, which is the minimum number of edges needed to connect u and v . For any $W \subseteq V$, let $\Gamma_G(W; k) = \{v \in V \mid \min_{w \in W} \text{dist}_G(w, v) = k\}$ denote the set of nodes of distance k from W in graph G , $k \geq 0$. Notice that $\Gamma_G(W; 0) = W$. A subgraph $G' = (V', E')$ of G is said *induced* if $E' = E \cap (V' \times V')$.

The *degree distribution* $P(k)$ is the proportion of nodes of degree k in a network, $k \geq 0$. Barabási and Albert [3] found that in many types of networks, including the social networks, $P(k)$ follows a power law, i.e.,

$$P(k) \sim k^{-\gamma} \quad (1)$$

for some constant $\gamma > 0$. This is called the *scale-free* phenomenon and has been widely applied to account for various behaviours [18, 26]. Barabási and Albert [3] also provided a BA model that can generate networks with this phenomenon.

The *clustering coefficient* is used to evaluate how nodes tend to knit together in a network. There are three kinds of them. The *local clustering coefficient* $\text{LCC}(v)$ for a node $v \in V$ is defined as 0 if $\deg(v) \leq 1$, otherwise ($\deg(v) \geq 2$)

$$\text{LCC}(v) = \frac{|\text{triangles to which } v \text{ belongs}|}{\deg(v) \times (\deg(v) - 1)/2}. \quad (2)$$

The *average clustering coefficient* $\text{ACC}(G)$ of a graph G is defined by

$$\text{ACC}(G) = \frac{\sum_{v \in V} \text{LCC}(v)}{n}. \quad (3)$$

And the *global clustering coefficient* $\text{GCC}(G)$ is defined by

$$\text{GCC}(G) = \frac{3 \times |\text{triangles in } G|}{|\text{open triplets in } G| + 3 \times |\text{triangles in } G|}, \quad (4)$$

where open triplet is an induced subgraph of G with three nodes and two edges. Clustering coefficients are widely used in studying social networks as social networks usually have greater clustering coefficients than random networks [29].

Another commonly used metric for studying social networks is the *average distance* $\bar{d}(G)$. It is well-known that the average distance between two people in the real world is small, known as the small-world phenomenon [15]. In network science, a *type* of networks is said small-world if

$$\bar{d}(G) \propto \log n \quad (5)$$

and $\text{GCC}(G)$ is not small (e.g., ≥ 0.2), as explained by the WS model [29]. Notice that this description cannot tell if a *given* network is small-world. Some studies [9, 19, 25] try to define if a given network is small-world by comparing it with random networks and/or lattice networks. Nonetheless, it is hard to say that these approaches are convincing since they rely on the definition of a so-called “equivalent” graph, which is not well-defined (thus there are various definitions).

Lastly, *core-periphery structure* is often considered in network studies [10, 13, 17, 23, 27, 28]. This term is intuitive and was initially employed to qualitatively describe the structural positions in a system and the dynamics among them (see, e.g., [10, 23]). A rigid and universally accepted definition, however, is still under debate [7]. The core structure may have different meanings in different studies. For instance, studies based on clique [4], group densities [11], and centrality [21], are employed under different assumptions.

3 The Proposed Pyramid Structure

We propose a definition of pyramid structure. Let $T = (V_T, E_T, r)$ denote a tree with node set V_T , edge set E_T and root r . It is called a *pyramid* if

$$1 < \rho_{\min} \leq \frac{|G_T(r; i+1)|}{|G_T(r; i)|} \leq \rho_{\max} \quad (6)$$

for some constants (called the *expansion ratios*) $\rho_{\min} \leq \rho_{\max}$, $i = 0, 1, \dots, h-1$, where h is the height of T . For simplicity, a singleton r is called a *trivial* pyramid of height 0. A pyramid $T = (V_T, E_T, r)$ is said *maximal* in a graph G if there is no other r -rooted pyramid $T' = (V_{T'}, E_{T'}, r)$ in G such that $V_{T'} \supsetneq V_T$.

For any graph and a node r , we can find a maximal r -rooted pyramid in G by a Breadth-First Search (BFS) started from r . Let T^* be the spanning tree found by the BFS. Let $h = \max\{k \geq 0 \mid |G_{T^*}(r; i)| > |G_{T^*}(r; i-1)|, i = 0, 1, \dots, k\}$. The partial tree T of T^* consisting of nodes of distance h and less from r is a maximal pyramid with $\rho_{\min} = \min\{|G_G(r; i)|/|G_G(r; i-1)|\} > 1$ and $\rho_{\max} = \max\{|G_G(r; i)|/|G_G(r; i-1)|\}$. Let us call such a pyramid a *basic* pyramid. We summarize the above argument as the following theorem.

Theorem 1. *A basic pyramid for an arbitrary root can be found in linear time.*

Based on the nature of BFS, we have the following Lemma.

Lemma 1. *There can be more than one basic pyramid with the same root. Nevertheless, they share the same height and the same set of nodes for each layer.*

The following arguments and empirical studies rely only on the set of nodes. Thus we do not differ basic pyramids of the same root for simplicity. See Fig. 2 for some examples.

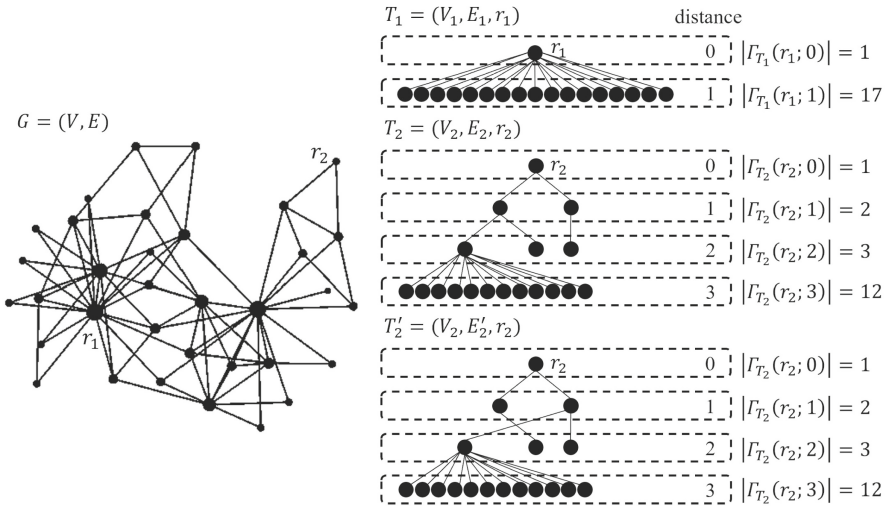


Fig. 2. An illustration of (basic) pyramids. Notice that T_2 and T_2' are of the same root, same node set, but different edge sets. (Source of the network: karate [22])

The proposed pyramid structure can provide alternative definitions of many metrics for networks. For instance, the degree distribution can be replaced by the size distribution of the height-1 maximal pyramids, since the degree of a node v is the size of the (unique) height-1 maximal pyramid of root v minus one (except for the degree-1 nodes where we do not need to subtract one).

For the clustering coefficients, we can view an open triplet a *minimal non-trivial* pyramid and vice versa. On the other hand, triangles can be viewed as an open triplet with a complemented edge. Depending on the complemented edge, we can view a triangle in exactly three different ways. Therefore, we can rewrite the definitions of the clustering coefficients using pyramids too.

Pyramid can also be used in studying the small-world phenomenon. Notice that the height h of a pyramid $T = (V_T, E_T, r)$ satisfies $h = O(\log |V_T|)$ by definition. Thus, for any two nodes $u, v \in V_T$, $\text{dist}_T(u, v) = O(\log |V_T|)$ since $\text{dist}_T(u, v) \leq 2h$. Therefore, if a network G has a large enough pyramid, we can expect that the average distance of G is $O(\log n)$. This connects the existence of large pyramid and the small-world phenomenon. We will further discuss it in the next section. Moreover, pyramid can be used to define a core-periphery structure of a given network. See the next section.

4 Empirical Studies and Applications

We study real social networks with the proposed pyramid structure. The networks were downloaded from the Network Repository site [22] of size not too small and not too large. As the result, we picked up 127 online social networks with 62 to 154,908 nodes. The list of them is provided in the appendix.

Let us call a pyramid in a graph G *large* if it consists of no less than $n/2$ nodes in G . This $1/2$ proportion can be decided depending on the application. For example, $2/3$ may be better in studying some representing issues.

4.1 On the Existence of Large Pyramid

We first study the large basic pyramids for each of 127 online social networks. By Lemma 1, we can use only one basic pyramid for each root. Thus, there are n basic pyramids in a graph of size n . Figure 3 shows the mean values of ρ_{\min} and ρ_{\max} and Fig. 4 shows the mean values and variances of the height for all large basic pyramids, respectively. Let

$$p_{\rho \geq 2}^* = \frac{\#\text{large basic pyramids with } \rho_{\min} \geq 2}{n}. \quad (7)$$

We observed that $p_{\rho \geq 2}^* > 0$ for all networks studied. Moreover, there are 123 networks with $p_{\rho \geq 2}^* > 0.2$, 113 with $p_{\rho \geq 2}^* > 0.5$, and 65 with $p_{\rho \geq 2}^* > 0.7$, showing that large pyramids are usually wide and common in a social network. Let

$$p_{h \leq 3}^* = \frac{\#\text{large basic pyramids with } h \leq 3}{n}. \quad (8)$$

We observed that there are 110 networks with $p_{h \leq 3}^* > 0.2$, 105 with $p_{h \leq 3}^* > 0.5$, and 104 with $p_{h \leq 3}^* > 0.7$, indicating that basic pyramids are usually short and large in a social network.

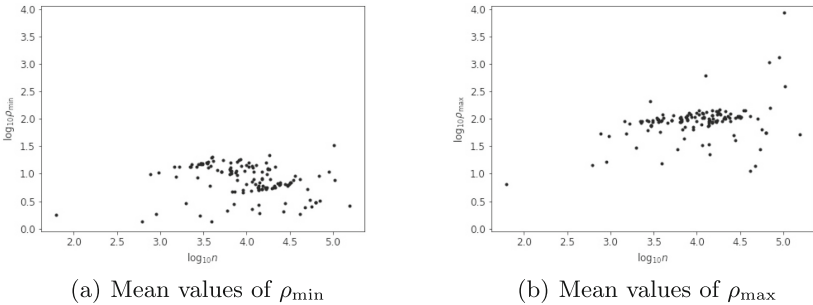


Fig. 3. Mean values of ρ_{\min} and ρ_{\max} for the 127 online social networks studied, where n denotes the number of nodes. Log-log scale is used with base 10.

This is surprising. It suggests that dominating (representing) a majority in a social network is much easier than what we thought, in the sense that it is highly

possible a small set of nodes to reach a majority of nodes in $O(\log n)$ hops. This phenomenon provides an evidence to the hypothesis by Zhao and Peng [30], who used it to provide an accurate estimation on the size of a parliament.

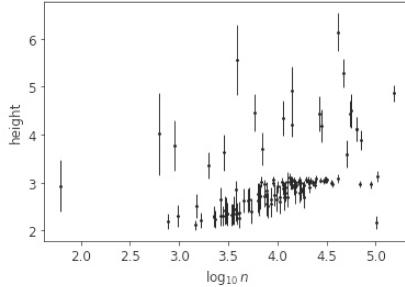


Fig. 4. Mean values (the y-axis value of points) and variances (the length of the lines) of heights for all large basic pyramids for the 127 online social networks studied, where n denotes the number of nodes. The logarithm scale on the x-axis is used with base 10.

We further study $p_{\rho \geq 2}^*$ for different types of networks. We studied four networks with almost the same number of nodes: **road-minnesota** is a road network ($n = 2,640$), **socfb-USFCA72** is an online social network ($n = 2,672$), a network generated by the WS model, and a network generated by the BA model. The latter two were generated by the generators `watts_strogatz_graph` ($n = 2672$, $k = 50$, $p = 0.3$) and `barabasi_albert_graph` ($n = 2672$, $m = 25$) in the NetworkX package of Python. The values of $p_{\rho \geq 2}^*$ were 0, 0.69, 0.17, and 0.90, respectively, different among different types of networks. An interesting observation was found when adjusting the probability p of rewiring edges in WS model. Adjusting p from 0.3 to 1.0, $p_{\rho \geq 2}^*$ increases as p increases, eventually becoming close to 1.

4.2 A Novel Structural Feature Revealed by Pyramid

In this subsection, we demonstrate a structure feature of networks that has not been known in the literature. Given a network with n nodes, let the normalized size of a pyramid T of size n_T be $\frac{n_T}{n}$. Figure 5 shows the distributions of the normalized size of basic pyramids of four networks including a road network **road-minnesota**, an online social network **socfb-USFCA72**, and two generated networks used in Subsect. 4.1.

It can be observed that in the road network **road-minnesota**, pyramids are all small (the normalized size is much less than 0.5), whereas in the social network **socfb-USFCA72**, most pyramids are large, and there is an obvious gap in the normalized size between small and large pyramids. Such a phenomenon can be observed in most of the real social networks studied (125 out of 127). On the other hand, all pyramids in WS model generated networks and BA model generated networks are large. This suggests, not surprisingly, that real social

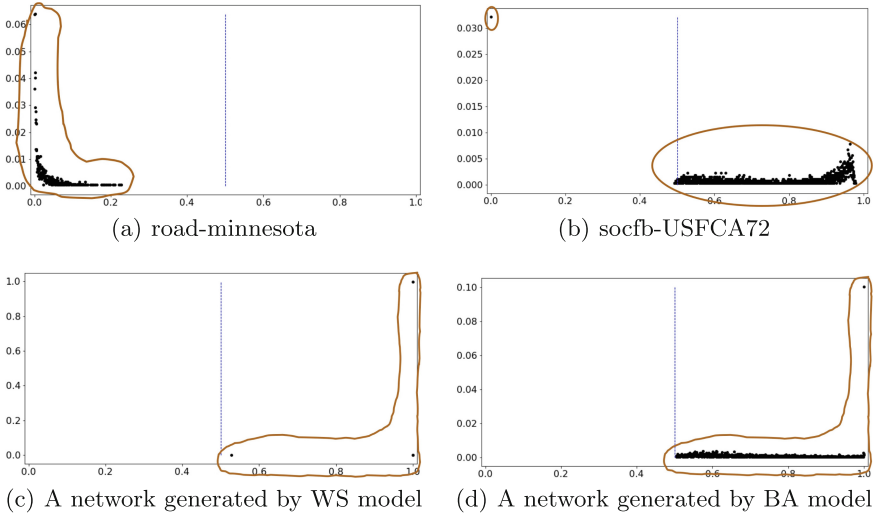


Fig. 5. Distributions of normalized sizes of basic pyramids of four networks.

networks usually mix connections by close location and location-independent virtual connections, for which existing models (the WS and the BA models) may fail to reproduce. Therefore, a new model is needed to study social networks.

4.3 Large Pyramid as a Core Structure

We first hypothesize that large pyramids contributes largely to the small-world phenomenon in social networks. Let T_B be a large pyramid with a node set V_B in a graph G with a set V of n nodes. As discussed in Sect. 3, every pair of nodes in T_B has distance $O(\log n)$, satisfying the most important feature of small-world networks. To study the clustering coefficients, we consider the two induced subgraphs G_B and $G_{\bar{B}}$ with node sets V_B and $V \setminus V_B$ respectively. Table 1 shows the results for two online social networks, where two large basic pyramids with $\rho_{\min} \geq 2$ were used. It can be observed that $\text{ACC}(G_B)$ is greater than 0.2 for both of the networks.

Table 1. Empirical results for two online social networks.

network	n	$\text{ACC}(G)$	$ V_B $	$\text{ACC}(G_B)$	$\text{ACC}(G_{\bar{B}})$
socfb-USFCA72	2,672	0.2668	1,337	0.2700	0.2503
socfb-Haverford76	1,446	0.3230	1,067	0.3088	0.3539

We can also observe a difference between the results for the two networks: $\text{ACC}(G_B) = \max\{\text{ACC}(G), \text{ACC}(G_B), \text{ACC}(G_{\bar{B}})\}$ holds for **socfb-USFCA72**

but not **socfb-Haverford76**. This suggests that structural importance of large pyramids may depend on the network. To address this issue and define a small-world network, we propose the next definition.

Definition (Small-World for a Single Network). A network G is said a (ρ, c) -small-world if there exists a large pyramid T_B in G such that $\rho_{\min} \geq \rho$ and $\text{ACC}(G_B) \geq c$ for some constants $\rho > 1, c > 0$.

We remark that this is a well-defined definition for any given network. Such a definition of small-world for a given network is significant [25].

Next, we use pyramid to define a core structure in the network as follows. Let T be a large basic pyramid in a network G with root r . Let $V_j = \bigcup_{0 \leq i \leq j} T(r; i)$ denote the set of nodes on T within distance j from $r, j \geq 0$. And let k be the minimum j such that all connected components of the subgraph G'_k induced by the node set $V \setminus V_j$ are small (i.e., with size less than $n/2$). We call the subgraph G_k induced by V_k a (connected) core of G . See an illustration in Fig. 6.

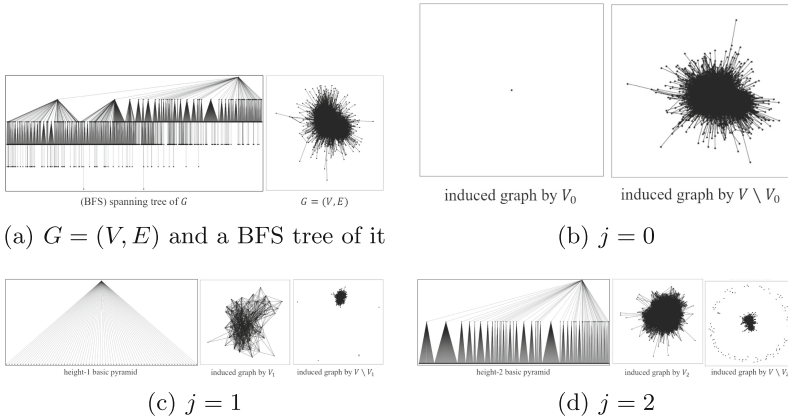
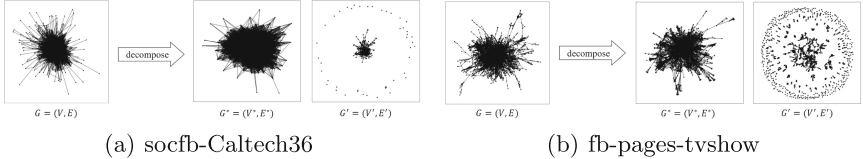


Fig. 6. A visualization of the proposed core structure for an online social network socfb-USFCA72 ($n = 2,672$), for which $k = 2$.

Table 2 shows clustering coefficients for the core defined above in the network **socfb-USFCA72**. It shows $\text{ACC}(G_j) = \max\{\text{ACC}(G), \text{ACC}(G_j), \text{ACC}(G'_j)\}$ for $j = 1, 2$, which suggests that G_j has strong inter-connection. Figure 7 visualizes two more instances, showing the core structure defined by this study is powerful.

Table 2. Clustering coefficients for Fig. 6, where $\text{ACC}(G) = 0.2668$.

k	$ V_k $	$\text{ACC}(G_k)$	$\text{ACC}(G'_k)$
0	1	0.0000	0.2667
1	71	0.6328	0.2675
2	1,337	0.2700	0.2503

**Fig. 7.** Visualization of the decomposition of two more online social networks.

5 Conclusion

Pyramid structures have been previously invoked in scholarly work to study representative nodes in social networks. However, a gap exists in the literature regarding both a precise definition and thorough empirical research on the topic. This paper aims to fill that void by offering a rigorous definition of pyramid structures. Our empirical studies, conducted across a wide array of real-world social networks, corroborate existing hypotheses about the prevalence of large pyramid structures in these networks.

Additionally, we identified a novel feature concerning pyramid sizes: they are generally small in road networks, large in networks generated by the WS and BA models, and variable in size within social networks. This observation underscores the need for a new, more adaptable model for studying real-world social networks.

Beyond contributing a clearer understanding of pyramid structures, our research holds implications for the broader field of network studies. We discussed how these insights could inform the design of network metrics and structures, touching upon elements like degree distribution, clustering coefficient, core-periphery structures, and the small-world phenomenon. Therefore, this study not only addresses a specific gap in the existing literature but also offers a foundation for more nuanced, effective approaches to network analysis in the future.

Acknowledgements. This research was funded by JSPS KAKENHI Grant Numbers 18K11182 and 23K10997. We thank the anonymous reviewers for their valuable comments.

Appendix

This appendix lists the names of networks studied in this study. People who are interested can download these networks by accessing the URLs which are in

the form of “<https://networkrepository.com/>” + name of network + “.php”. For example, one can download the data of **fb-pages-artist** by accessing <https://networkrepository.com/fb-pages-artist.php>.

fb-pages-artist	socfb-Columbia2	socfb-OR	socfb-UGA50
fb-pages-company	socfb-Cornell5	socfb-Penn94	socfb-UIllinois
fb-pages-food	socfb-Dartmouth6	socfb-Pepperdine86	socfb-UMass92
fb-pages-government	socfb-Duke14	socfb-Princeton12	socfb-UNC28
fb-pages-media	socfb-Emory27	socfb-Reed98	socfb-UPenn7
fb-pages-politician	socfb-FSU53	socfb-Rice31	socfb-USC35
fb-pages-public-figure	socfb-Georgetown15	socfb-Rochester38	socfb-USF51
fb-pages-sport	socfb-GWU54	socfb-Rutgers89	socfb-USFCA72
fb-pages-tvshow	socfb-Hamilton46	socfb-Santa74	socfb-UVA16
soc-anybeat	socfb-Harvard1	socfb-Simmons81	socfb-Vanderbilt48
soc-BlogCatalog	socfb-Haverford76	socfb-Smith60	socfb-Vassar85
soc-brightkite	socfb-Howard90	socfb-Stanford3	socfb-Vermont70
soc-buzznet	socfb-Indiana	socfb-Swarthmore42	socfb-Villanova62
soc-dolphins	socfb-JMU79	socfb-Syracuse56	socfb-Virginia63
soc-douban	socfb-JohnsHopkins55	socfb-Temple83	socfb-Wake73
soc-epinions	socfb-Lehigh96	socfb-Tennessee95	socfb-WashU32
socfb-American75	socfb-Maine59	socfb-Texas80	socfb-Wellesley22
socfb-Amherst41	socfb-Maryland58	socfb-Texas84	socfb-Wesleyan43
socfb-Auburn71	socfb-Mich67	socfb-Trinity100	socfb-William77
socfb-Baylor93	socfb-Michigan23	socfb-Tufts18	socfb-Williams40
socfb-BC17	socfb-Middlebury45	socfb-Tulane29	socfb-Wisconsin87
socfb-Berkeley13	socfb-Mississippi66	socfb-UC33	socfb-wosn-friends
socfb-Bingham82	socfb-MIT	socfb-UC61	socfb-Yale4
socfb-Bowdoin47	socfb-MSU24	socfb-UC64	soc-gemsec-HR
socfb-Brandeis99	socfb-MU78	socfb-UCF52	soc-gemsec-HU
socfb-Brown11	socfb-nips-ego	socfb-UChicago30	soc-gemsec-RO
socfb-BU10	socfb-Northeastern19	socfb-UCLA	soc-hamsterster
socfb-Bucknell39	socfb-Northwestern25	socfb-UConn	soc-LiveMocha
socfb-Cal65	socfb-NotreDame57	socfb-UCSB37	soc-slashdot
socfb-Caltech36	socfb-NYU9	socfb-UCSC68	soc-themarker
socfb-CMU	socfb-Oberlin44	socfb-UCSD34	soc-wiki-Vote
socfb-Colgate88	socfb-Oklahoma97	socfb-UF	karate
road-minnesota			

References

1. Barabási, A.L.: Scale-free networks: a decade and beyond. *Science* **325**(5939), 412–413 (2009)
2. Barabási, A.L.: *Network Science*. Cambridge University Press, Cambridge (2016)
3. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999)
4. Borgatti, S.P., Everett, M.G.: Models of core/periphery structures. *Soc. Networks* **21**(4), 375–395 (2000)
5. Bowler, W.M., Brass, D.J.: Relational correlates of interpersonal citizenship behavior: a social network perspective. *J. Appl. Psychol.* **91**(1), 70–82 (2006)
6. Czégel, D., Palla, G.: Random walk hierarchy measure: what is more hierarchical, a chain, a tree or a star? *Sci. Rep.* **5**(1), 1–14 (2015)
7. Gallagher, R. J., Young, J. G., Welles, B. F.: A clarified typology of core-periphery structure in networks. *Sci. Adv.* **7**, eabc9800 (2021)
8. Granovetter, M.: The impact of social structure on economic outcomes. *J. Econ. Perspect.* **19**(1), 33–50 (2005)
9. Humphries, M.D., Gurney, K.: Network “small-world-ness”: a quantitative method for determining canonical network equivalence. *PLoS ONE* **3**(4), e0002051 (2008)
10. Laumann, E.O., Pappi, F.U.: *Networks of Collective Action: A Perspective on Community Influence Systems*. Academic Press, New York (1976)
11. Lee, S.H., Cucuringu, M., Porter, M.A.: Density-based and transport-based core-periphery structures in networks. *Phys. Rev. E* **89**, 032810 (2014)
12. Lu, C., Yu, J. X., Li, R. -H., Wei, H.: Exploring hierarchies in online social networks. In: *IEEE Trans. Knowl. Data Eng.* **28**(8), 2086–2100 (2016). <https://doi.org/10.1109/TKDE.2016.2546243>
13. Luo, F., Li, B., Wan, X.-F., Scheuermann, R.H.: Core and periphery structures in protein interaction networks. *BMC Bioinform.* **10**(4), 1–11 (2009)
14. Luo, J., Magee, C.L.: Detecting evolving patterns of self-organizing networks by flow hierarchy measurement. *Complexity* **16**(6), 53–61 (2011)
15. Milgram, S.: The small world problem. *Psychol. Today.* **2**(1), 60–67 (1967)
16. Mones, E., Vicsek, L., Vicsek, T.: Hierarchy measure for complex networks. *PLoS ONE* **7**(3), e33799 (2012)
17. Mullins, N. C., Hargens, L. L., Hecht, P. K., Kick, E. L.: The group structure of cocitation clusters: a comparative study. *Am. Sociol. Rev.* 552–562 (1977)
18. Nan, Y., Wenying, L., Wei, G.: Study on scale-free characteristic on propagation of cascading failures in power grid. *IEEE 2011 EnergyTech*, 1–5 (2011)
19. Neal, Z.P.: How small is it? Comparing indices of small worldliness. *Netw. Sci. (Camb. Univ. Press).* **5**(1), 30–44 (2017)
20. Papachristos, A.V.: Murder by structure: dominance relations and the social structure of gang homicide. *Am. J. Sociol.* **115**(1), 74–128 (2009)
21. Rombach, M.P., Porter, M.A., Fowler, J.H., Mucha, P.J.: Core-periphery structure in networks. *SIAM J. Appl. Math.* **74**(1), 167–190 (2014)
22. Rossi, R. A., Ahmed, N. K.: The network data repository with interactive graph analytics and visualization. *AAAI* (2015). <https://networkrepository.com>
23. Snyder, D., Kick, E.L.: Structural position in the world system and economic growth, 1955–1970: A multiple-network analysis of transnational interactions. *Am. J. Sociol.* **84**(5), 1096–1126 (1979)
24. Taagepera, R.: The size of national assemblies. *Soc. Sci. Res.* **1**(4), 385–401 (1972). [https://doi.org/10.1016/0049-089X\(72\)90084-1](https://doi.org/10.1016/0049-089X(72)90084-1)

25. Telesford, Q.K., Joyce, K.E., Hayasaka, S., Burdette, J.H., Laurienti, P.J.: The ubiquity of small-world networks. *Brain Connect.* **1**(5), 367–375 (2011)
26. Teschendorff, A.E., Banerji, C.R., Severini, S., Kuehn, R., Sollich, P.: Increased signaling entropy in cancer requires the scale-free property of proteininteraction networks. *Sci. Rep.* **5**(1), 1–9 (2015)
27. Verma, T., Russmann, F., Araújo, N., Nagler, J., Herrmann, H.: Emergence of core-peripheries in networks. *Nat. Commun.* **7**, 10441 (2016)
28. Villeseche, F., Sinani, E.: From presence to influence: gender, nationality and network centrality of corporate directors. *Work Employ Soc.* **37**(2), 486–504 (2023)
29. Watts, D.J., Strogatz, S.H.: Collective dynamics of “small-world” networks. *Nature* **393**(6684), 440–442 (1998)
30. Zhao, L., Peng, T.: An allometric scaling for the number of representative nodes in social networks. In: Masuda, N., Goh, K.I., Jia, T., Yamanoi, J., Sayama, H. (eds) *Proc. NetSci-X*, pp. 49–59 (2020)



Dual Communities Characterize Structural Patterns and Robustness in Leaf Venation Networks

Philipp C. Böttcher¹, Franz Kaiser², Henrik Ronellenfitsch³,
Vito Latora^{4,5,6,7}, and Dirk Witthaut^{1,2}

¹ Institute of Energy and Climate Research – Energy Systems Engineering (IEK-10),
Forschungszentrum Jülich, 52428 Jülich, Germany

{p.boettcher,d.witthaut}@fz-juelich.de

² Institute for Theoretical Physics, University of Cologne, 50937 Köln, Germany

³ Physics Department, Williams College, Williamstown, MA 01267, USA

⁴ School of Mathematical Sciences, Queen Mary University of London, London, UK

⁵ Department of Physics and Astronomy, University of Catania, 95125 Catania, Italy

⁶ INFN Sezione di Catania, Via S. Sofia, 64, 95125 Catania, Italy

⁷ Complexity Science Hub Vienna, 1080 Vienna, Austria

Abstract. The structural organization of supply networks plays an important role in their functioning. In this work, we discuss how standard community detection analysis can be complemented by the definition of communities in the networks' plane dual. Such communities are characterized not by weak but by strong connectivity between the different components of the network. We extract dual communities in leaf venation networks, finding that they can reveal structural features not visible to traditional community detection methods. Furthermore, we show that the presence of dual community structures suppresses failure spreading and that dual communities can be used to classify different leaf types.

Keywords: Graph Duality · Supply Networks · Leaf Venation Networks · Community Detection

1 Introduction

Understanding the structural organization of a network is a central aspect of network science [17]. On a mesoscopic scale, the presence of community structures is essential for the functioning and stability of a network. Typically, communities are defined as groups of nodes that have a higher connectivity within each group than between the groups [22, 24]. Due to this different connectivity, the existence of community structures generally reduces the spreading of information [21] or diseases [25]. Furthermore, the presence of communities strongly affects the dynamic stability of a system [18, 20] and its robustness to failures [32].

Many man-made and biological networks are spatially embedded and planar [1]. For each plane graph, i.e., each planar drawing of a planar graph, it is possible to define its *dual graph* as follows. Every node of the dual graph corresponds to a face of the original graph's planar drawing (see Fig. 1). If two faces

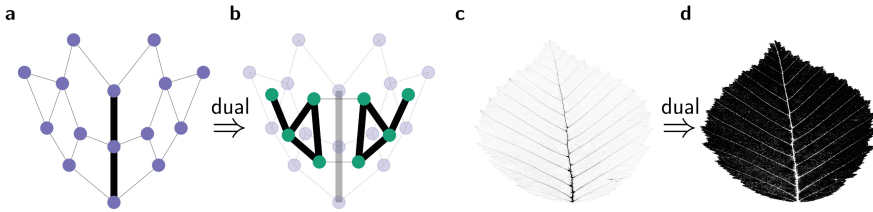


Fig. 1. Primal and dual graphs a,b, Construction of the dual of a plane graph. Every face of the original graph corresponds to one node of the dual graph. Two dual nodes are connected if the facets share an edge. Primal and dual weights are reciprocal according to Eq. (6). **c,d**, The venation network of a beech leaf and its plane dual.

share an edge, their corresponding dual nodes are connected by a dual edge. Dual graphs turn out to be very useful in the analysis of flow networks [3], in networked dynamical systems [19], and also in the design of network algorithms [13, 28, 29].

In this article, we will show that dual graphs can be very useful in revealing patterns in the structure of networks that are undetected by a simple representation in terms of primal (i.e., original) graphs. In particular, we will review a method introduced in Ref. [14] to define and study *dual communities* in supply networks. Subsequently, we focus on the analysis of a particular case of biological supply networks, namely leaf venation networks. We will show how dual communities are closely related to the hierarchical organization of the network and allow us to distinguish between leaf venation networks of different leaf species.

2 Dual Graphs of Weighted Spatial Networks

Graph duality was first introduced in the geometric study of Platonic solids and has found numerous applications. Given a plane graph $G(V, E)$, i.e., a graph drawn in the plane without any edges crossing, consisting of a set of vertices V and edges E , the dual G^* is defined by the following geometric procedure. Every face of the primal graph constitutes a vertex $v^* \in V^*$ of the dual graph. Two dual vertices, i.e., two faces, are adjacent if they share at least one primal (i.e., original) edge.

Real spatial networks are often strongly heterogeneous, which is incorporated by assigning weights w_e to the edges $e \in E$. For example, links in transportation networks may strongly differ in their capacity. We must thus extend the definition of the dual graph to obtain a meaningful dual representation of a spatial network. The key question is: What is the weight of a dual edge and how is it related to the weights of the primal (i.e., original) edges?

We base our analysis on two characteristic matrices defining the structure of a graph, the Laplacian $\mathbf{L} \in \mathbb{R}^{N \times N}$ and weighted edge-node incidence matrix $\mathbf{I} \in \mathbb{R}^{M \times N}$ where $N = |V|$ and $M = |E|$. Notably, the Laplacian is also used in

spectral graph bisection, which is a classic method to determine the community structure of a graph [9]. The elements of the Laplacian are given by [23]

$$L_{ij} = \begin{cases} -w_{ij} & \text{if } i \text{ is connected to } j, \\ \sum_{(i,k) \in E} w_{ik} & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $w_{ij} > 0$ is the weight of an edge (i, j) . To define the edge-node incidence matrix we label the graph's edges as $\ell = 1, 2, \dots, M$ and fix an orientation for each edge. The matrix elements of \mathbf{I} are given by

$$I_{\ell n} = \begin{cases} \sqrt{w_{\ell}} & \text{if line } \ell \text{ starts at node } n, \\ -\sqrt{w_{\ell}} & \text{if line } \ell \text{ ends at node } n, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Note that the incidence matrix defined in this way coincides with the gradient in discrete calculus [6]. In terms of the incidence matrix, the Laplacian reads

$$\mathbf{L} = \mathbf{I}^{\top} \mathbf{I} \in \mathbb{R}^{N \times N}, \quad (3)$$

where the symbol \top denotes the matrix transpose.

We now transfer these concepts to the dual graph G^* . To this end, we generalize the initial geometric definition to a more general mathematical framework. A formal abstract definition is provided by algebraic graph theory, based on the duality between a graph's cut space and its cycle space [5]. The fundamental elements are cycle flows, i.e., flows that do not have a source or sink. The cycle flows form a vector space of dimension $N^* = M - N + 1$ which is equivalent to the kernel of the incidence matrix \mathbf{I} . This equivalence formalizes the property that cycle flows have no source or sink. We can now find a basis of N^* fundamental cycles and summarize them in the edge-cycle incidence matrix $\mathbf{C} \in \mathbb{R}^{M \times N^*}$ which satisfies the fundamental relation

$$\mathbf{I}^{\top} \mathbf{C} = \mathbf{0}. \quad (4)$$

In a plane graph, a distinguished basis is given by the N^* faces of the graph. The edge-cycle incidence matrix $\mathbf{C} \in \mathbb{R}^{M \times N^*}$ is then given by

$$C_{\ell c} = \begin{cases} 1/\sqrt{w_{\ell}} & \text{if edge } \ell \text{ is part of face } c, \\ -1/\sqrt{w_{\ell}} & \text{if reversed edge } \ell \text{ is part of face } c, \\ 0 & \text{otherwise.} \end{cases}$$

The edge-cycle incidence matrix \mathbf{C} encodes the structure of the dual graph in the same way as the edge-node incidence matrix \mathbf{I} encodes the structure of the primal graph. We can thus define the Laplacian matrix of the dual graph in an analogue way as

$$\mathbf{L}^* = \mathbf{C}^{\top} \mathbf{C} \in \mathbb{R}^{N^* \times N^*}. \quad (5)$$

One thus finds that edges in the dual graph are weighted with the inverse of edge weights in the primal graph. More precisely, two dual nodes, i.e., two faces c and d , are linked by a dual edge with weight

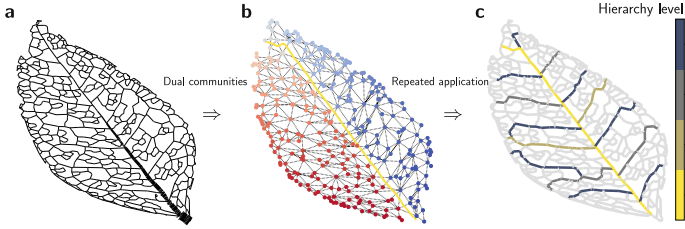


Fig. 2. Dual communities by repeated spectral graph bisection. **a**, A leaf venation network of *Bursera hollickii*. The thickness of the lines indicates the conductivity w_{ij} . The dual network and the dual Laplacian \mathbf{L}^* are computed as described in Sect. 2. **b**, Dual nodes are assigned to a community according to the value of the dual Fiedler vector, cf. Eq. (7). The value $(\mathbf{v}_2^*)_i$ is indicated by colors (blue = positive, red = negative). The yellow line shows the detected community boundary. **c**, The process is iterated to detect community boundaries at different hierarchical levels.

$$w_{c,d}^* = \sum_{\ell \in c,d} \frac{1}{w_\ell}, \quad (6)$$

where the sum runs over all edges ℓ that are shared by the faces c and d . We emphasize that primal and dual weights are reciprocal, which forms the basis of our further analysis. An alternative derivation is given in [14].

3 Communities and Hierarchies in Dual Graphs

The dual of a spatial network can be analyzed in the same way as the original primal network and, as we will now show, provides additional insights that are hidden in the primal graph.

We focus on community structures in biological supply networks, more precisely in leaf venation networks. These are always plane networks, hence their dual graphs can be computed as described above. We use a data set from Ref. [27], which has been obtained from high-resolution photographs. The data set that we used for classifying leaf venation networks consists of 40 leaves that are classified into 6 species. Edges correspond to veins of the leaf, while nodes are defined as the joints of the respective veins. The weight w_{ij} corresponds to the conductivity of a vein (i, j) . Assuming a laminar flow of water through an edge, the conductivity scales as with the radius r_{ij} as $w_{ij} \propto r_{ij}^4$ according to the Hagen-Poiseuille law [4].

To extract the dual communities, we apply two standard algorithms for community detection to the dual network. First, we use spectral graph bisection [9], as it is based on the graph Laplacian, which plays a central role in the definition of dual graphs. The eigenvalues of a Laplacian of an undirected graph are always

real and non-negative and can be ordered as $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$. The second eigenvalue λ_2 provides a measure of the algebraic connectivity of the graph and the associated eigenvector \mathbf{v}_2 , also referred to as Fiedler vector, can be used to decompose a network into two communities [8,9]. A vertex i is assigned to community 1 or 2 according to the respective entry of the Fiedler vector \mathbf{v}_2 ,

$$\text{sign}(\mathbf{v}_2)_i \begin{cases} > m \Rightarrow i \text{ belongs to community 1} \\ < m \Rightarrow i \text{ belongs to community 2.} \end{cases} \quad (7)$$

This process is iterated by splitting the network into components (i.e., communities) and applying spectral graph bisection separately to each part. An example of this procedure is shown in Fig. 2. Given the original network, we first compute the dual Laplacian \mathbf{L}^* and its Fiedler vector \mathbf{v}_2^* as shown in the figure. We then iterate this procedure and map the community boundaries.

As a second method, we use the Louvain method for community detection [2] to the dual network, which uses a combination of Refs. [2,34]. In short, the algorithm starts by assigning every node to a different isolated community. Subsequently, each isolated single node i is assigned to its neighboring community which leads to the maximal increase in modularity given by

$$\Delta Q = \frac{k_{i,\text{in}}}{2m} - \xi \frac{\mathcal{S}_{\text{tot}} \cdot k_i}{2m^2} \quad (8)$$

where m is the total number of edges in the graph, k_i is the weighted degree of node i , $k_{i,\text{in}}$ is the sum of the weights of the edges connecting node i to the community \mathcal{C} , \mathcal{S}_{tot} is the sum of the weights of all edges connecting to nodes in \mathcal{C} , and ξ is the resolution parameter that decides whether larger or smaller communities are favored. The procedure is repeated until no further increase in modularity can be achieved. In the following step, the discovered communities are interpreted as nodes and the procedure is run iteratively. The entire procedure is terminated if no increase in modularity above a small threshold is achieved. The Louvain method is a standard benchmark algorithm for community detection. We here use the implementation in Python's NetworkX package [11] with a small resolution parameter, i.e., $\xi = 0.01$ for the examples presented in Fig. 3, to obtain larger communities.

For comparison, the results of the application of the two algorithms to the dual graphs are shown in Fig. 3 together with those obtained for the original primal graphs. A clear, interpretable community structure does not emerge from the primal graphs. In the case of the Maple leaf, we may speculate that the communities detected by the Louvain method correspond to the area mostly served by one of the five strong veins emerging from the root. However, this interpretation does not generalize to the case of the Beech leaf. Furthermore, we find that spectral graph bisection decomposition and the Louvain method do not yield consistent results.

In contrast, we observe a consistent and easily interpretable community structure in the dual graphs. The boundaries between the detected communities mostly correspond to the primary and secondary veins of the leaf. Hence, dual

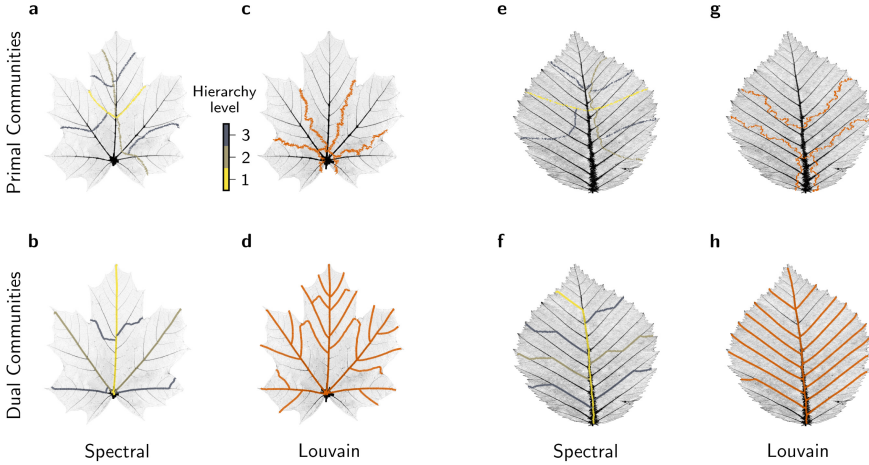


Fig. 3. Primal and dual communities in leaf venation networks. Community boundaries in the leaf venation network of a Maple (i.e., *Acer platanoides*) and a Beech (i.e., *Corylus avellana*) leaf, respectively, as obtained by repeated spectral bisection (colored lines in panels a, b, e and f) and by modularity maximization using the Louvain method for community detection with resolution parameter $\xi = 0.01$ (orange lines in panels c, d, g and h). Upper row: Primal graphs do not exhibit a clear, interpretable community structure. Lower row: The community boundaries of the dual graph clearly correspond to the primary and secondary veins of the leaf.

communities readily reveal the hierarchical organization of leaf venation networks. We find that Louvain method for community detection is especially suited to identify the essential primary and secondary veins. Spectral graph bisection is not as precise, but provides further information about the hierarchical organization, as different hierarchies are detected at successive steps of the algorithm. These examples show that duality can provide a new, valuable perspective on spatially embedded networks. We further underpin this finding in the following by proposing a method for the classification of leaf venation networks.

4 Classification of Leaf Venation Patterns

The main elements of leaf venation networks are readily identified by community detection algorithms applied to the dual graph as discussed above. Further important properties of the network can be obtained from a more detailed analysis of the dual Laplacian \mathbf{L}^* . First, the dual algebraic connectivity λ_2^* provides a measure of the strength of the central vein. It thus provides an important characteristic associated with the first level of the hierarchical network. Second, the Fiedler vectors \mathbf{v}_2^* encode structural information which may be used to classify different types of leaves.

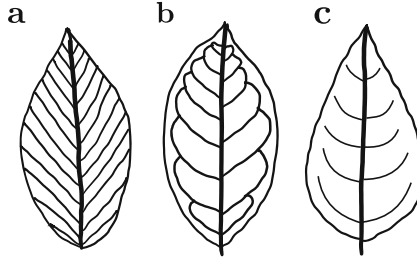


Fig. 4. Three classes of leaf venation networks according to Hickey [12]. **a**, Craspedodromous: Secondary veins extend to the boundary. **b**, Brochidodromous: Secondary veins join. **c**, Eucamptodromous: Secondary veins terminate within the leaf without joining.

Hickey introduced an influential classification scheme of leaf venation networks [12,30]. Focusing on secondary veins, three different types of leaves are sketched in Fig. 4. Secondary veins extend to the boundary for the Craspedodromous type. They join for the Brochidodromous type, and they terminate within the leaf for the Eucamptodromous type. Remarkably, the dual Fiedler vector shows rather clearly whether secondary veins join or terminate and thus enables the classification of leaf venation networks. We base our analysis on the second iteration of the spectral bisection method, as the first iteration is dominated by the primal vein. Consider for example a leaf of the species *Carpinus betulus*. We see in Fig. 5c,d that the secondary veins do not join but extend to the boundary. The areas between the secondary veins are identified by different values of the dual Fiedler vector \mathbf{v}_2^* , encoded by different colors in the figure. However, the differences diminish at the boundary of the leaf where secondary veins get weaker. A different picture is observed for leaves of species *Bursera simaruba*. (see Fig. 5g,h) or *Protium sp. nov. 8* (see Fig. 6a,b). Secondary veins join and we observe extended areas with almost constant values of the $(\mathbf{v}_2^*)_i$. Plotting these values in ascending order reveals pronounced plateaus, which is not observed for *Carpinus betulus* (see Fig. 6d,e).

As shown above, the dual Fiedler vector can provide valuable information on the type of leaf venation networks. Hence, we propose the following classification scheme for leaf venation networks based on the second splitting obtained by repeated spectral bisection. We first calculate the dual Fiedler vector $\mathbf{v}_2^{*,a}$ for each community a and normalize the results to enable a comparison between different species. To this end, we apply an affine-linear mapping to all vector elements such that $\max_i(\mathbf{v}_2^*)_i = +1$ and $\min_i(\mathbf{v}_2^*)_i = -1$. Finally, we sort the entries of the vector in ascending order and evaluate the difference $(\Delta\mathbf{v}_2^*)_i = (\mathbf{v}_2^*)_i - (\mathbf{v}_2^*)_{i-1}$ between subsequent changes. The distribution of these changes $p(\Delta\mathbf{v}_2^*)$ is an indicator of the amount and flatness of plateaus of the dual Fiedler vector and hence of its structure: We expect a network with plateaus that are more strongly pronounced to yield larger jumps in the dual Fiedler such that

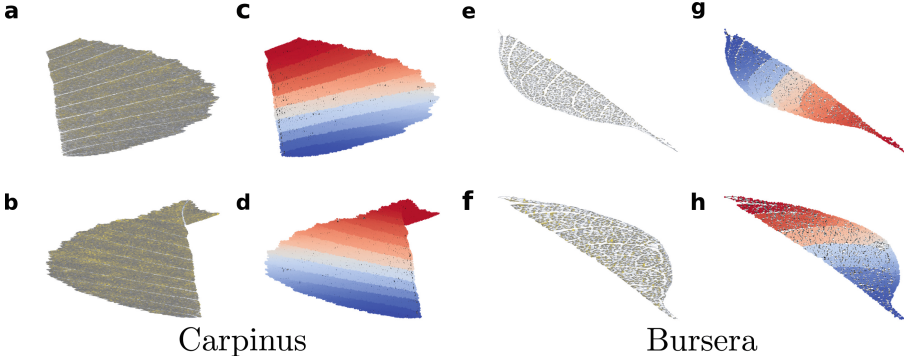


Fig. 5. Secondary veins and dual Fiedler vectors for two species of leaves.

First, we separate both species into the two halves that were identified during the first spectral clustering step. The resulting dual networks can be seen in panels **a,b** and **e,f** for *Carpinus betulus* and *Bursera simaruba*, respectively. Second, we focus on the results of the second iteration of the spectral bisection by evaluating the dual Fiedler vector and plotting its elements $(\mathbf{v}_2^*)_i$, which is shown encoded by the color (blue/red) in panels **c,d** and **g,h**. One clearly observes different areas separated by the secondary veins. For *Carpinus betulus* secondary veins extend to the boundary of the leaf (Craspedodromous type), but get weaker at the boundary where areas join smoothly. For *Bursera simaruba*, secondary veins join (Brochidodromous type).

the probability of observing large jumps, $p_{\text{jump}} = p(\Delta \mathbf{v}_2^* > 10 \cdot \text{median}(\Delta \mathbf{v}_2^*))$, encodes this structural feature.

We evaluate the two structural features λ_2^* and $p(\Delta \mathbf{v}_2^* > 10 \cdot \text{median}(\Delta \mathbf{v}_2^*))$ for various samples of leaves from different species. We find that each species covers a characteristic area in feature space (see Fig. 6f). For instance, *Carpinus* leaves are characterized by low values of p_{jump} and medium values of λ_2^* . The secondary veins extend to the boundary (Craspedodromous type) but they are rather weak. Hence, the dual Fiedler spectrum is rather smooth, and p_{jump} is small. *Bursera* leaves are characterized by high values of both λ_2^* and p_{jump} , indicating strong primary and secondary veins.

5 Dual Communities and Leaf Robustness

Dual community structures strongly affect a network's robustness to edge failures [10, 14]. For leaf venation networks, damages may be caused by pathogens, insects, or herbivores [16, 30]. Here we study the impact of a single edge failure on network flows. A common model for venation networks assumes a laminar flow through each vein (i, j) . According to the Hagen-Poiseuille law, we thus find

$$F_{i \rightarrow j} = w_{ij}(p_i - p_j), \quad (9)$$

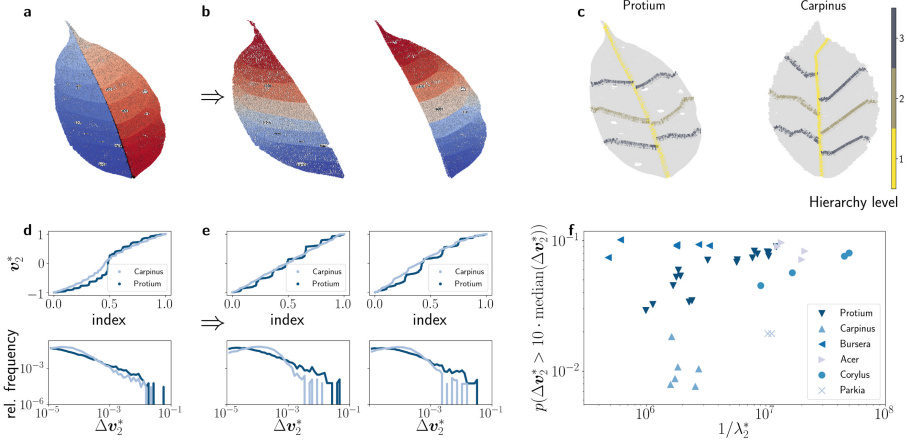


Fig. 6. Dual communities reveal the hierarchical organization of leaf venation networks. **a,b**, Repeated spectral bisection in the dual graph of a *Protium sp. nov.* 8 leaf using the Fiedler vector v_2^* (color code) yields a hierarchical decomposition of the network. **c**, Hierarchical levels reveal the structural differences between different leaf genera. **d,e**, The differences between genera are captured by the sorted dual Fiedler vector v_2^* (top) and the distribution of its spacings $(\Delta v_2^*)_i = (v_2^*)_i - (v_2^*)_{i-1}$ (bottom). **f**, The distribution of spacings along with the dual Fiedler value λ_2^* allow for a classification of leaves of different genera. The evaluation of the distribution is based on the dual Fiedler vector of the second level of decomposition.

where the conductivity of a vein scales with the radius as $w_{ij} \propto r_{ij}^4$. The flows must further satisfy the continuity equation at every joint such that

$$\sum_j F_{i \rightarrow j} = F_i^{\text{in}}. \quad (10)$$

For a leaf, we have an inflow $F_i^{\text{in}} > 0$ only for a single node representing the root of the primary vein. All other nodes consume water and nutrients and thus have $F_i^{\text{in}} < 0$. Fixing the in- and outflows F_i^{in} , the two Eqs. (9) and (10) completely determine the pressures and flows in the network. We note that the flows satisfy the same fundamental equations as electric currents in resistor networks and we can apply the same mathematical tools for their analysis.

We now examine the impact of the failure of a single vein (r, s) . To this end, we evaluate the flow change on every vein (n, m) after the failure, normalized by the initial flow of the failing vein,

$$\text{LODF}_{(n,m),(r,s)} = \frac{F_{nm}^{\text{after}} - F_{nm}^{\text{before}}}{F_{rs}^{\text{before}}}. \quad (11)$$

We note that this ratio is an important quantity in the monitoring of electric power grids, where it is commonly referred to as a line outage distribution factor (LODF) [28].

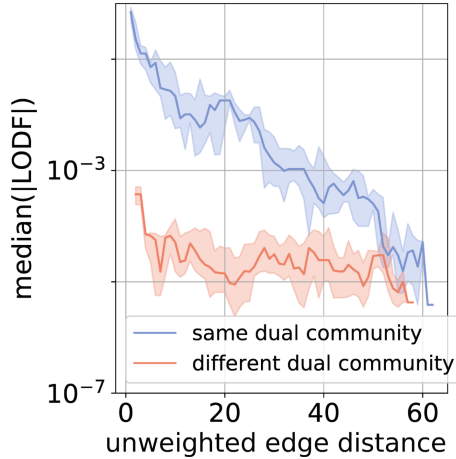


Fig. 7. Dual communities determine the robustness to edge failures. We consider the damage of a single vein in a leaf of *Bursera hollickii* (cf. Fig. 3) and evaluate the normalized flow change in another vein after the damage according to Eq. (11). The figure shows the median $M_{s,o}(d)$ over all pairs of veins at a given distance d that are either in the same (s) or the other (o) dual community with the shaded regions indicating the 25% to 75% quantile. The impact of the failure is strongly suppressed in the other dual community, $M_o(d) \ll M_s(d)$, especially at small distance d .

To quantify the effect of the community boundary, we consider all pairs of veins in the leaf (cf. Ref. [15]). We must take into account that LODFs generally decrease with distance [33]. Thus, we take the median over all pairs at a fixed unweighted distance d ,

$$M_{s,o}(d) = \underset{\text{dist}[(n,m),(r,s)] = d}{\text{median}} |\text{LODF}_{(n,m),(r,s)}|. \quad (12)$$

When evaluating the median, we take into account only pairs of veins that are either in the same dual community (subscript s) or in the other community (subscript o) as the failing link.

We observe that the dual community structure has a strong impact on the flow changes [14] resulting in the flow changes between communities being suppressed by order of magnitude for small to medium distances (see Fig. 7). In other words, the central vein ‘shields’ the flow changes [10]. Remarkably, we find that $M_s(d)$ decreases strongly with the distance d as expected, while $M_o(d)$ shows only a weak dependence on d . Hence the difference of $M_s(d)$ and $M_o(d)$ diminishes for very large distances.

6 Conclusion

In this work, we have used the concept of plane graph duality to study leaf venation networks. Detecting community structures in both the original (i.e., primal)

graph and the corresponding dual graph, we have shown that dual communities not only correspond to leaf components separated by primary and secondary veins but also have important implications for the spreading of disturbances. While primal communities are separated by weak mutual connectivity, dual communities are separated by strong connectivity along the community boundary. Both types of community boundaries provide a “shielding” effect. That is, the impact of a failure remains contained in the respective community. Furthermore, the community structure identified in the dual graph can be used to classify different types of leaf venation networks, which emphasizes the benefit of analyzing the dual in tandem with the original primal graph.

Note that dual communities also play an important role in other supply networks such as electric power transmission grids [14]. It has been shown that the emerging community structure depends on the fluctuations of in- and outflows, which strongly increase during the transition to renewable power generation. The concept of dual communities is *not* restricted to a certain algorithm – any community detection algorithm can be applied to the dual graph. In particular, one can readily extend the analysis to overlapping [7] or hierarchical [26,31] communities. We conclude that graph duality is a valuable tool in the analysis of spatially embedded networks across disciplines.

References

1. Barthélemy, M.: Spatial networks. *Phys. Rep.* **499**(1), 1–101 (2011). <https://doi.org/10.1016/j.physrep.2010.11.002>
2. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**(10), P10,008 (2008). <https://doi.org/10.1088/1742-5468/2008/10/P10008>
3. Bollobás, B.: Modern graph theory. No. 184 in Graduate texts in mathematics. Springer (1998). <https://doi.org/10.1007/978-1-4612-0619-4>
4. Coomes, D.A., Heathcote, S., Godfrey, E.R., Shepherd, J.J., Sack, L.: Scaling of xylem vessels and veins within the leaves of oak species. *Biol. Lett.* **4**(3), 302–306 (2008). <https://doi.org/10.1098/rsbl.2008.0094>
5. Diestel, R.: *Graph Theory*. Springer, New York (2010). <https://doi.org/10.1007/978-3-662-53622-3>
6. Elmoataz, A., Lezoray, O., Boughleux, S.: Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. *IEEE Trans. Image Process.* **17**(7), 1047–1060 (2008). <https://doi.org/10.1109/TIP.2008.924284>
7. Evans, T.S., Lambiotte, R.: Line graphs, link partitions, and overlapping communities. *Phys. Rev. E* **80**, 016,105 (2009). <https://doi.org/10.1103/PhysRevE.80.016105>
8. Fiedler, M.: Algebraic connectivity of graphs. *Czechoslov. Math. J.* **23**(2), 298–305 (1973)
9. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3), 75–174 (2010). <https://doi.org/10.1016/j.physrep.2009.11.002>
10. Gavrilchenko, T., Katifori, E.: Resilience in hierarchical fluid flow networks. *Phys. Rev. E* **99**(1), 012,321 (2019). <https://doi.org/10.1103/PhysRevE.99.012321>

11. Hagberg, A., Swart, P., S Chult, D.: Exploring network structure, dynamics, and function using networkx. Tech. rep., Los Alamos National Lab. (LANL), Los Alamos, NM (United States) (2008)
12. Hickey, L.J.: A revised classification of the architecture of dicotyledonous leaves. In: M. CR, C. L (eds.) *Anatomy of the Dicotyledons*. Vol. 1. Systematic Anatomy of Leaf and Stem, pp. 25–39. Clarendon Press, Oxford (1979)
13. Hörsch, J., Ronellenfitsch, H., Witthaut, D., Brown, T.: Linear optimal power flow using cycle flows. *Electric Power Syst. Res.* **158**, 126–135 (2018). <https://doi.org/10.1016/j.epsr.2017.12.034>
14. Kaiser, F., Böttcher, P.C., Ronellenfitsch, H., Latora, V., Witthaut, D.: Dual communities in spatial networks. *Nat. Commun.* **13**(1), 7479 (2022). <https://doi.org/10.1038/s41467-022-34939-6>
15. Kaiser, F., Latora, V., Witthaut, D.: Network isolators inhibit failure spreading in complex networks. *Nat. Commun.* **12**(1), 3143 (2021). <https://doi.org/10.1038/s41467-021-23292-9>
16. Katifori, E., Szöllösi, G.J., Magnasco, M.O.: Damage and fluctuations induce loops in optimal transport networks. *Phys. Rev. Lett.* **104**, 048,704 (2010). <https://doi.org/10.1103/PhysRevLett.104.048704>
17. Latora, V., Nicosia, V., Russo, G.: *Complex networks: principles, methods and applications*, first edn. Cambridge University Press (2017)
18. Manik, D., Rohden, M., Ronellenfitsch, H., Zhang, X., Hallerberg, S., Witthaut, D., Timme, M.: Network susceptibilities: Theory and applications. *Physical Review E* **95**(1) (2017). <https://doi.org/10.1103/PhysRevE.95.012319>
19. Manik, D., Timme, M., Witthaut, D.: Cycle flows and multistability in oscillatory networks. *Chaos: An Interdisciplinary J. Nonlinear Sci.* **27**(8), 083,123 (2017). <https://doi.org/10.1063/1.4994177>
20. May, R.M.: Will a large complex system be stable? *Nature* **238**(5364), 413 (1972). <https://doi.org/10.1038/238413a0>
21. Nematzadeh, A., Ferrara, E., Flammini, A., Ahn, Y.Y.: Optimal Network Modularity for Information Diffusion. *Phys. Rev. Lett.* **113**(8) (2014). <https://doi.org/10.1103/PhysRevLett.113.088701>
22. Newman, M.E.J.: Modularity and community structure in networks. *Proc. Natl. Acad. Sci. U.S.A.* **103**(23), 8577–8582 (2006). <https://doi.org/10.1073/pnas.0601602103>
23. Newman, M.E.J.: *Networks: An Introduction*. Oxford University Press (2010)
24. Newman, M.E.J.: Communities, modules and large-scale structure in networks. *Nat. Phys.* **8**(1), 25–31 (2012). <https://doi.org/10.1038/nphys2162>
25. Pastor-Satorras, R., Castellano, C., Van Mieghem, P., Vespignani, A.: Epidemic processes in complex networks. *Rev. Mod. Phys.* **87**(3), 925 (2015). <https://doi.org/10.1103/RevModPhys.87.925>
26. Peixoto, T.P.: Hierarchical block structures and high-resolution model selection in large networks. *Phys. Rev. X* **4**(1), 011,047 (2014)
27. Ronellenfitsch, H., Lasser, J., Daly, D.C., Katifori, E.: Topological phenotypes constitute a new dimension in the phenotypic space of leaf venation networks. *PLOS Comput. Biol.* **11**(12), e1004,680 (2015). <https://doi.org/10.1371/journal.pcbi.1004680>
28. Ronellenfitsch, H., Manik, D., Hörsch, J., Brown, T., Witthaut, D.: Dual theory of transmission line outages. *IEEE Trans. Power Syst.* **32**, 4060–4068 (2017). <https://doi.org/10.1109/TPWRS.2017.2658022>

29. Ronellenfitsch, H., Timme, M., Witthaut, D.: A dual method for computing power transfer distribution factors. *IEEE Trans. Power Syst.* **32**(2), 1007–1015 (2017). <https://doi.org/10.1109/TPWRS.2016.2589464>
30. Roth-Nebelsick, A., Uhl, D., Mosbrugger, V., Kerp, H.: Evolution and function of leaf venation architecture: a review. *Ann. Bot.* **87**(5), 553–566 (2001). <https://doi.org/10.1006/anbo.2001.1391>
31. Schaub, M.T., Li, J., Peel, L.: Hierarchical community structure in networks. *Phys. Rev. E* **107**, 054,305 (2023). <https://doi.org/10.1103/PhysRevE.107.054305>
32. Shekhtman, L.M., Shai, S., Havlin, S.: Resilience of networks formed of interdependent modular networks. *New J. Phys.* **17**(12), 123,007 (2015). <https://doi.org/10.1088/1367-2630/17/12/123007>
33. Strake, J., Kaiser, F., Basiri, F., Ronellenfitsch, H., Witthaut, D.: Non-local impact of link failures in linear flow networks. *New J. Phys.* **21**(5), 053,009 (2019). <https://doi.org/10.1088/1367-2630/ab13ba>
34. Traag, V.A., Waltman, L., Van Eck, N.J.: From louvain to leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**(1), 1–12 (2019). <https://doi.org/10.1038/s41598-019-41695-z>



Tailoring Benchmark Graphs to Real-World Networks for Improved Prediction of Community Detection Performance

Catherine Schwartz^{1,2(✉)}, Cetin Savkli¹, Amanda Galante¹,
and Wojciech Czaja²

¹ Johns Hopkins University Applied Physics Laboratory, Laurel, MD, USA

² University of Maryland Department of Mathematics, College Park, MD, USA

catherine.schwartz@jhuapl.edu

Abstract. Analysts interested in understanding the community structure of a particular real-world network will often simply choose a popular community detection algorithm and trust the generated results without further investigation, but algorithm performance can vary depending on the network characteristics. We demonstrate that by running experiments on benchmark graphs tailored to match characteristics of a real-world network of interest, a better understanding can be obtained on how community detection algorithms will perform on the real-world network. We show that the correlation between the performance of the community detection methods on a publicly available dataset to the average performance of the same methods on the corresponding tailored benchmark graphs is high whereas the correlation with LFR benchmark graphs is negative. This means the methods that performed well on the tailored graphs also performed well on the real-world network but methods that perform well on LFR graphs did not perform well on the real-world network, demonstrating that the proposed methodology has merit.

Keywords: community detection · benchmark graphs · network models

1 Introduction

A network is a graph - a structure containing pairwise relationships between objects - that can serve as a representation of a real-world system. Networks are often thought to have community structure meaning the objects in the network can be easily partitioned into groups, called communities, that are more densely connected within than between [8]. Most networks, such as people connected by friendships in social networks or webpages connected by hyperlinks on the Web, do not have meaningful ground truth communities, though knowing its community structure may provide valuable information to an analyst [21]. Therefore, communities need to be detected with algorithms. A wide range of algorithms have been proposed to detect communities but there is no one method

that performs best on all networks, known as the No Free Lunch Theorem [21]. A common approach used to compare the performance of community detection methods is to measure their ability to detect ground truth communities in benchmark graphs [11, 16, 17, 30]. Studies that use this approach typically use general parameters that create realistic community structure then provide guidance on how to choose an appropriate method based on the performance results [30]. Unfortunately, analysts may follow the guidance even though the real-world network they are working with may not have any resemblance to the benchmark graphs used in the study. Even for a specific benchmark model, algorithms can perform differently (e.g., in sparse vs highly connected conditions). Additionally, each study we have found has focused on a single type of community structure without guidance on how to determine if a given network may have that type of structure. We demonstrate that by running experiments on tailored benchmark graphs, a better understanding can be obtained on how well community detection methods will work on a specific real-world network of interest.

To create tailored benchmark graphs, analysts should understand different structural characteristics associated with their real-world network to help them choose appropriate input parameters for a benchmark model. The most common benchmark model with community structure is the LFR benchmark [11], but there exist real-world networks that have community structure that the LFR benchmark cannot create. Therefore, analysts should also consider a newer benchmark model called the nPSO benchmark [17] that can generate community structure that is different than what the LFR benchmark can create. When parameters are chosen for each model that attempt to match the real-world network as best as possible, the analyst can explore how similar the resulting graphs are to their real-world network and choose the best one to use for experiments. A publicly available dataset is used to demonstrate the methodology.

The remainder of the paper is organized as follows. In Sect. 2 we introduce two benchmark models with community structure. Section 3 introduces the publicly available dataset analyzed and Sect. 4 demonstrates how we created tailored benchmark graphs for the dataset and how we determined which were a better fit. Section 5 contains the results of the community detection experiments run on the tailored benchmark graphs and how they compare to the community detection performance on the real-world network. Section 6 contains our conclusion and discussion.

2 Background

2.1 LFR Benchmark Graphs

Proposed by Lancichinetti, Fortunato, and Radicchi in 2008, the *LFR benchmark* is a planted partition model that allows for different numbers of vertices with degree distributions and community size distributions that can be right-skewed as observed in many real-world networks [11]. To generate *LFR graphs* from the LFR benchmark, users can specify the number of vertices N , the average degree $\langle k \rangle$, the maximum degree k_{max} , a degree distribution scaling parameter

γ , a community size distribution scaling parameter β , the minimum and maximum community sizes denoted as c_{min} and c_{max} respectively, and the mixing parameter μ . An example LFR graph can be seen in Fig. 1a. While the resulting LFR graphs contain community structure, it does not explain the mechanisms in which community structure may come about.

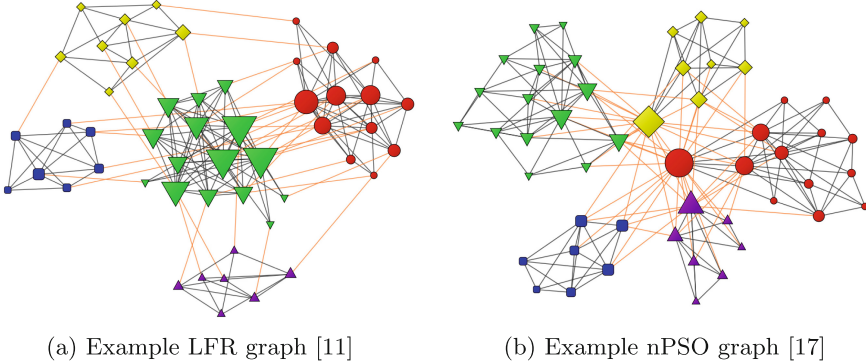


Fig. 1. Example graphs with 5 communities. Vertex color/shape indicate community. Vertex size indicates degree. Internal vs external edges denoted by color.

2.2 nPSO Benchmark Graphs

The *non-uniform Popularity Similarity Optimization (nPSO) benchmark* is a network growth model proposed by Muscoloni and Cannistraci that does provide an explanation to the mechanisms in which community structure may about [17]. The concept of popularity is the main force behind the preferential attachment mechanism – newer vertices added to a model prefer to connect to more popular vertices with higher degree than less popular vertices with lower degree [1]. Papadopoulos et al. introduced another concept called similarity, which can be modeled using a measure like cosine similarity, to determine the angular distance of vertices randomly placed on a circle [20]. Muscoloni and Cannistraci added non-uniformity in the angular coordinates to generate community structure specified by a user. To generate *nPSO graphs*, users can specify the number of vertices N , a parameter equal to half the average degree of a vertex $m = \langle k \rangle / 2$, a degree distribution scaling parameter γ , a temperature parameter $T \in [0, 1)$ used to tune the mixing, along with the means in radians, standard deviations in radians, and mixing proportions of their desired communities. With these parameters, the user is able to specify the number of communities and relative expected sizes of those communities. An example nPSO graph can be seen in Fig. 1b.

2.3 Related Work

While there have been a large number of studies on the performance of community detection algorithms on LFR graphs [2, 7, 10, 14, 30, 31], less research has

been conducted on studying the performance of community detection algorithms on nPSO graphs, largely due to the fact that the nPSO benchmark was proposed more recently. In the paper that introduced this new benchmark [17], an initial glimpse at the detectability of its ground truth communities was conducted by analyzing the NMI of the detected communities using the Louvain method for a variety of parameter settings. A follow-up study was published the same year by Muscoloni and Cannistraci where the performance of the Louvain method was compared to Infomap, Walktrap, and Label Prop in a variety of parameter settings [16]. No studies were found that analyzed more than one benchmark model at a time.

There are common sets of parameters used to generally evaluate community detection algorithms [10, 11, 16, 17, 30, 31]. Few authors attempt to match benchmark parameters to a specific real-world network they are studying [2, 19, 27] and those that do not compare the performance on the benchmark graphs to the performance on the corresponding real-world network. Community detection algorithms can perform quite differently with different parameter settings for the same network model (e.g., high vs low characteristic path length), so it can be important to understand how similar a real-world network of interest is to benchmark graphs used to validate the methods they are considering using.

3 The Real-World Network

SNAP datasets (the Stanford Large Network Dataset Collection) [12] contains a large number of publicly available networks, but only one network has non-overlapping ground truth communities, specifically, the *email-Eu-core* network has 1,005 vertices, 25,571 edges, and 42 non-overlapping communities of varying sizes. Each vertex in the email-Eu-core network represents a member of a research institution and an edge exists if the two members sent an email back and forth to each other over an 18-month period of time. The ground truth communities correspond to the department an individual member belongs to. The nature of the dataset makes it a good candidate for community analysis because it seems plausible that the department structure of the research institution would lead to groups in the network that are more densely connected within than between [21] and determining department structure from email traffic may be a use case an analyst would be interested in investigating. While the original network is directed with self-loops, for community detection purposes a simple graph version, G_{EU} , is made resulting in 1,005 vertices and 16,064 edges. Additional graph-level metrics are the average degree: 32, minimum degree: 0, maximum degree: 345, assortativity: -0.026 , characteristic path length: 2.584, average local clustering coefficient: 0.450, global clustering coefficient: 0.267, rich-clubness (p-value): 0.000, number of communities: 42, minimum community size: 1, and maximum community size: 42. A cumulative degree distribution plot, along with a hop plot [6] and a rich-clubness plot [15] are also helpful to determine the appropriate parameters for tailored benchmark graphs.

4 The Real-World Network Compared to Tailored Benchmark Graphs

The goal of creating benchmark graphs similar to G_{EU} is not to find graphs that match G_{EU} exactly. For example, the fact that G_{EU} has 1,005 vertices does not need to be matched exactly.

An attempt is first made to create LFR graphs similar to G_{EU} with an emphasis placed on obtaining the most similar maximum degree vertices with the most similar number of communities. To attempt to generate LFR graphs similar to G_{EU} , called LFR_EU, the parameters $N = 1,000$, $\langle k \rangle = 32$, $k_{max} = 345$, $\gamma = 2$, $\beta = 1.5$, $c_{min} = 3$, $c_{max} = 109$, and $\mu = 0.55$ were chosen. With LFR graphs, differing numbers of communities are made for each realization. Because a high degree vertex cannot be placed in a community unless its internal degree is less than the community size [11], it is difficult for this particular set of parameters to create the smaller community sizes that would be needed to get up to 42 communities. $\gamma = 2$ is used, yielding an instance that contained 41 communities whereas using $\gamma = 3$ only ever produced a much smaller number of communities. Out of an ensemble of 1,000 LFR graphs with the stated parameters used, the graph with the greatest number of communities and highest degree is considered the best LFR graph, G_{LFR} , for the sole purpose of creating plots and metrics for comparison to G_{EU} .

Unlike LFR graphs, nPSO graphs allow the user to set the number of communities but the maximum degree of a graph is determined by the growth model and is dependent on parameters m and γ . To obtain nPSO graphs similar to G_{EU} , called nPSO_EU, the parameters were set to $N = 1,000$, $m = 16$, and 42 equidistant means around the circle are set with standard deviations equal to $\frac{1}{6} * \frac{2\pi}{42}$ (* = scalar multiplication). The mixing proportions are used to create unequal sized communities, specifically, the proportions are set using a power law with exponent $\beta = 2.6$. $T = 0.3$ is used to create similar clustering and $\gamma = 3$ for the nPSO graphs so that maximum degree in each benchmark graph is similar to G_{EU} whereas setting $\gamma = 2$ creates vertices with much larger maximum degrees. Out of an ensemble of 1,000 nPSO graphs with the stated parameters used, the graph with the lowest maximum degree is considered the best nPSO graph, G_{nPSO} , for the sole purpose of creating plots and metrics for comparison to G_{EU} .

While a number of graph-level metrics similar to G_{EU} can be created by either type of benchmark model, G_{nPSO} is much more similar to G_{EU} than G_{LFR} is. A major difference is seen in the characteristics of the communities, as seen in Fig. 2 (all plots and metrics are not shown due to space limitations). The conductance (the ratio between the external and total degree of a community [7]) of each community in G_{LFR} are quite uniform by design [11], whereas in the real-world network G_{EU} these values can vary quite significantly. Low degree vertices are often connected to high degree vertices outside of their community in G_{EU} , leading to high conductance values for smaller communities. This is expected in networks with hierarchical structure [25]. nPSO benchmark graphs are able to generate graphs with hierarchical community structure (see Fig. 2c) whereas

LFR benchmark graphs cannot (see Fig. 2b), leading to community structure much more similar to certain real-world networks (see Fig. 2a). This implies that the benchmark nPSO_EU is a potentially good benchmark for G_{EU} , whereas LFR_EU is not.

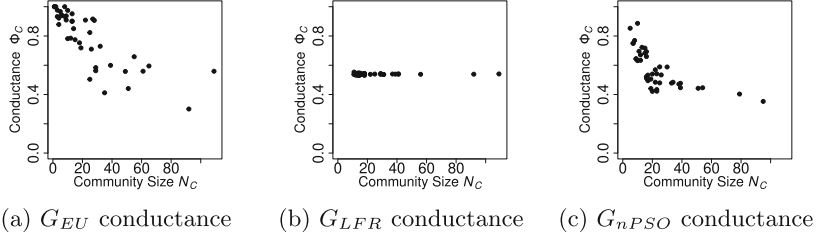


Fig. 2. Conductance plots for the simplified email-Eu-core network, G_{EU} , and for the best LFR graph, G_{LFR} , and best nPSO graph, G_{nPSO} .

nPSO graphs do not have community structure similar to the conventional paradigm [1]. Modularity is a measure often used to assess a partitioning of a network into communities where, for an unweighted network, the fraction of edges that fall within the communities is compared to what would be expected in a randomized degree-preserving multigraph [4]. Small communities with high conductance have a large fraction of their edges falling outside their communities – if a network is unweighted, these communities may negatively contribute to the modularity score. Modularity maximization is one of the most common techniques in density centric community detection, used by the popular Louvain method [3]. By properly weighting the edges in a way that there are higher weights within the communities and lower weights between communities, it is possible to still utilize traditional community detection algorithms.

5 Results

As stated in the previous section, without proper weighting, community detection algorithms that maximize modularity are at risk of not being able to resolve smaller communities with high conductance, as seen in nPSO graphs. Pre-weighting graphs is a technique that has been used to solve a number of existing problems with community detection algorithms, such as offsetting the resolution limit [2] and mitigating problems with extreme degeneracy [10]. The set of pre-weighting methods analyzed in this paper can be seen in Table 1. Note that the InvPrLinkCo method was slightly modified to include only 2-hop and 3-hop weights, combined evenly. Just 2-hop and just 3-hop weights for the link cohesion metrics is denoted by appending a `_S2` or `_S3` to the names, respectively.

The main community detection algorithm used for experiments will be the Louvain method [3], but other community detection methods are used to understand the benefits of pre-weighting with InvPrLinkCo - the best performing

Table 1. Pre-weighting methods used for community detection. ‘°’ = cycle-based.

Name	Type	Description	Paper
Modified ECC	Local°	Modified edge clustering coefficient	[23]
CN Ratio	Local	Percentage of common neighbor vertices	[9]
RA2	Local	A repulsion attraction rule	[18]
NeighborCo	Semi-Local°	Neighborhood coherence	[2]
SimRank3	Semi-Local	Truncated SimRank after 3 iterations	[31]
InvPrLinkCo	Semi-Local°	Inversely Proportional Link Cohesion	[27]
LoCoLinkCo	Semi-Local°	Local Community Link Cohesion	[29]
Inverse EB	Global	Normalized inverse edge betweenness	[10]
Khadivi et al	Global	‘Inverse EB’ multiplied by ‘CN Ratio’	[10]
RNBRW	Global°	Renewal non-backtracking random walks	[14]

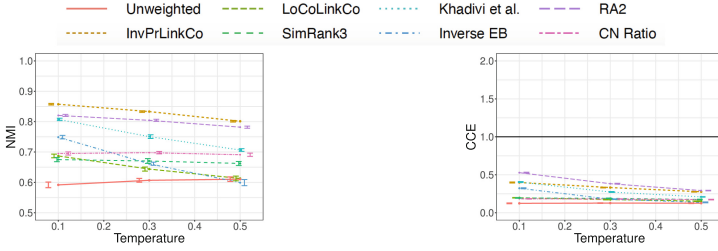
pre-weighting method for the Louvain method - for a wider range of algorithms. These algorithms include Fast Greedy [5], Infomap [26], Walktrap [22], Label Prop [24], and HSBM [13].

All experiments will utilize the nPSO_EU benchmark built to be similar to the G_{EU} graph, described in Sect. 4. Because ground truth is available for the benchmark graphs and G_{EU} , the results are presented as normalized mutual information abbreviated as *NMI* [7] and community count error abbreviated as *CCE* where $CCE = \bar{C}/C$ (the ratio of the average number of detected communities over the average number of real communities), as used in [30], both with desired values equaling 1. Results are summarized from 30 random repetitions. When comparing against the baseline methods, mean values are plotted with 95% confidence intervals, assuming normal distributions. When comparing against cycle-based methods, boxplots are shown.

5.1 Experiments with the Louvain Method

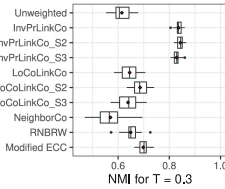
Because some of the cycle-based methods are more computationally expensive, experiments are run on a single temperature, $T = 0.3$, instead of varying the value, as is done with the other baseline methods. The community detection results for nPSO_EU are in Fig. 3, which shows InvPrLinkCo obtaining higher NMI than the other baseline pre-weighting methods, both cycle-based and non-cycle-based. The same experiments were run against other real-world parameter settings with nPSO graphs, such as sparser settings, and InvPrLinkCo also obtained higher NMI than the other pre-weighting methods in those cases.

Using the InvPrLinkCo_S2 weights by themselves also works well on the nPSO benchmark graphs with parameters chosen to be similar to specific real-world networks, likely because there tends to be a larger number of smaller communities and a few larger communities [11]. The InvPrLinkCo_S2 weights can be computed faster than the combined weights, so in certain situations,

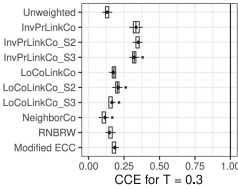


(a) NMI for baseline methods

(b) CCE for baseline methods



(c) NMI for cycle-based methods



(d) CCE for cycle-based methods

Fig. 3. Community detection results for nPSO_EU using the Louvain method with different pre-weighting methods. The unweighted graph is also used as a reference. (a) and (b) report the NMI and CCE for non-cycle-based methods. (c) and (d) report for $T = 0.3$ the boxplots of NMI and CCE for cycle-based methods.

such as when the characteristic path length is relatively low, it may be beneficial to use just the S2 weights instead of both.

All of the methods undercount the number of true communities in nPSO_EU due to the high average degree and large number of small community sizes, as expected due to the resolution limit [2]. The next best performing pre-weighting methods are RA2 [18] and the measure proposed by Khadivi et al. [9, 10].

InvPrLinkCo creates cycle-based weights that are better than the other cycle-based baselines (see Fig. 3c) which cannot differentiate between triangles and rectangles among high degree vertices created due to their popularity and triangles and rectangles among low degree vertices tied together due to their similarity (see Fig. 3d), demonstrating that taking the degree of the vertices along nearby alternate paths into account can be beneficial.

5.2 Experiments with Other Community Detection Algorithms

InvPrLinkCo is not specific to the Louvain method and has the potential to improve the communities detected by other algorithms. Therefore, Figs. 4 and 5 show results from experiments testing the potential performance improvements from using InvPrLinkCo to pre-weight graphs for the other community detection methods.

Pre-weighting the highly connected nPSO_EU graphs with InvPrLinkCo provide higher mean NMI values (see Fig. 4) and more accurate predictions on

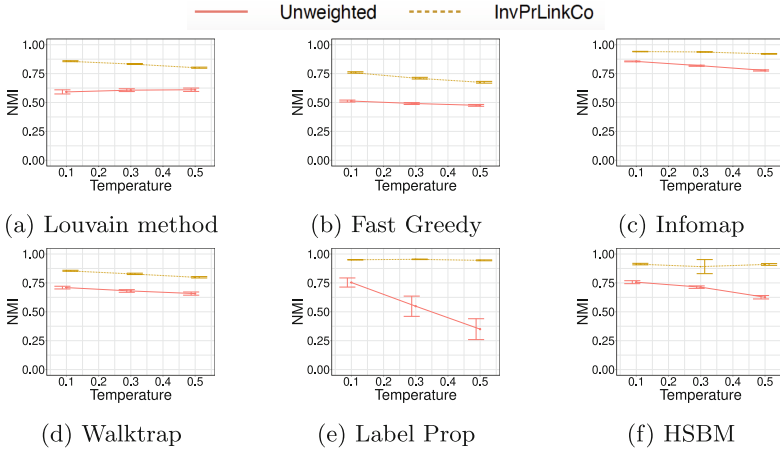


Fig. 4. NMI for InvPrLinkCo versus unweighted on nPSO_EU.

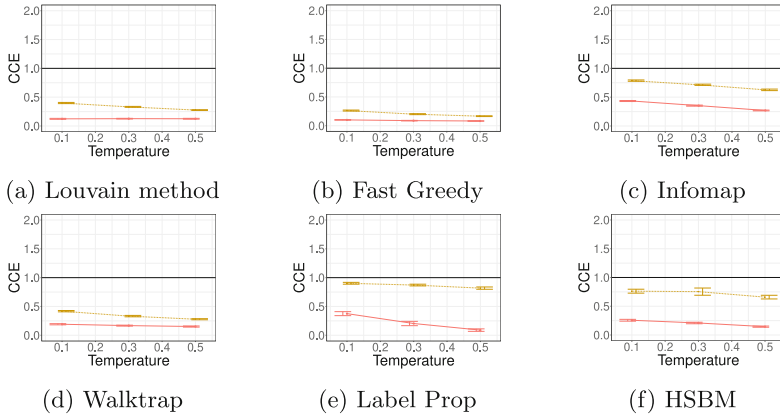


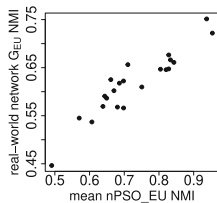
Fig. 5. CCE for InvPrLinkCo versus unweighted on nPSO_EU.

the number of communities (see Fig. 5) for all community detection methods tested. This was not necessarily true in all of the parameter settings tested for nPSO graphs. While pre-weighting nPSO graphs with InvPrLinkCo for modularity maximations algorithms (Louvain and Fast Greedy) improved results across all parameter settings tested, methods like Infomap and Label Prop sometimes degraded in performance with the added weights. A hypothesis for why this occurred is that these methods are able to utilize cycles of higher order into their solution on unweighted graphs, therefore focusing on only shorter cycles when the characteristic path length is relatively high can be a problem.

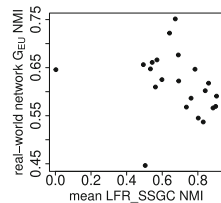
5.3 Comparing the Performance on Tailored Benchmark Graphs to the Performance on the Real-World Network

Now that it is understood how pre-weighting affects the performance of community detection methods on the tailored benchmark graphs, the performance is measured on the real-world network. The performance of all the community detection algorithms with and without pre-weighting using InvPrLinkCo and all baseline pre-weighting methods with the Louvain method was measured on their ability to predict the ground truth communities of G_{EU} , corresponding to departments an individual belongs to. The NMI value of each approach was compared to the mean NMI value obtained on nPSO_EU when $T = 0.3$ for that approach, with results shown in Fig. 6a. Three methods are left off of the plot, specifically the pre-weighted and unweighted HSBM which threw an exception probably due to the isolated vertices in the network and the unweighted Label Prop which obtained an NMI of 0.03 due to placing all of the vertices in the largest connected component into a single community.

While the NMI values on the real-world network G_{EU} are consistently lower than the mean NMI on the tailored benchmark graphs, the linear correlation between the two sets of values is 0.93 across all of the community detection methods that provided reasonable results. The two top performing approaches switch ordering, with Infomap pre-weighted with InvPrLinkCo obtaining an NMI of 0.75 and Label Prop pre-weighted with InvPrLinkCo obtaining an NMI of 0.71. Walktrap pre-weighted with InvPrLinkCo obtains the third highest NMI of 0.68, the Louvain method pre-weighted with InvPrLinkCo obtains the fourth highest NMI of 0.67, and the Louvain method pre-weighted with InvPrLinkCo_S2 obtains the fifth highest NMI of 0.66.



(a) nPSO_EU from Section 4
(tailored benchmark graphs)



(b) LFR_SSGC from [27]
(unrelated benchmark graphs)

Fig. 6. Comparison of the mean NMI from benchmark graphs to the NMI from G_{EU} of different the community detection approaches.

In contrast, Fig. 6b compares the NMI value from G_{EU} of each approach included in Fig. 6a to the mean NMI value obtained on the LFR benchmark graphs from [27] with $\langle k \rangle = 30$ and $\mu = 0.6$, which we will refer to as LFR_SSGC. Here, the linear correlation is -0.25. Infomap on the unweighted LFR_SSGC has a mean NMI of 0.003, whereas it has an NMI among the top 50% on G_{EU} . The

top performing approach for LFR_SSGC is the Louvain method pre-weighted with LoCoLinkCo, which yields an NMI among the bottom third on G_{EU} .

6 Conclusion and Discussion

The strong correlation between the performance of community detection methods on the real-world network to the average performance of the same methods on the corresponding tailored benchmark graphs demonstrates that the proposed methodology has merit. In an upcoming paper, this methodology is also applied to another publicly available dataset that is assumed to have community structure more similar to the LFR benchmark. While there was no ground truth associated with that dataset, different characteristics of the performance results were shown to be similar between the tailored benchmark graphs and the real-world network. A tool was also created to help determine what type of community structure a network may have independent of any ground truth. In another study, the tool was used to demonstrate that sparse retweet networks are similar to the nPSO benchmark and yield very different characteristics than typical nPSO graphs (regarding rich-clubness). The methods that performed best on the tailored benchmark graphs for the email-Eu-core network were among the worst performing algorithms on the tailored benchmark graphs for the retweet networks even though both were generated from the nPSO benchmark, providing further evidence that creating tailored benchmark graphs are important.

Not all networks will have community structure that closely matches one of the benchmark models described in this paper. That said, understanding the similarities and differences a specific real-world network has to the benchmarks may help an analyst decide on what community detection methods to use. This paper has focused on finding clusters that maximize internal density while minimizing outside connections [28], which may not be the type of community an analyst needs for their specific problem.

Acknowledgments. This paper was written under a Janney Grant from Johns Hopkins University Applied Physics Laboratory. We also thank Carlo V. Cannistraci, who introduced the authors to the nPSO benchmark at the 8th International Conference on Complex Networks and their Applications in 2019.

References

1. Barabási, A.: Network science. *Phil. Trans. R. Soc. A* **371**(1987) (2013)
2. Berry, J.W., Hendrickson, B., LaViolette, R.A., Phillips, C.A.: Tolerating the community detection resolution limit with edge weighting. *Phys. Rev. E* **83**(5) (2011)
3. Blondel, V.D., Guillaume, J., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech.: Theory Exp.* **2008**(10) (2008)
4. Bollobás, B.: A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *Eur. J. Comb.* **1**(4) (1980)
5. Clauset, A., Newman, M., Moore, C.: Finding community structure in very large networks. *Phys. Rev. E* **70**(6) (2004)

6. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. *ACM SIGCOMM CCR* **29**(4) (1999)
7. Fortunato, S., Hric, D.: Community detection in networks. *Phys. Rep.* **659** (2016)
8. Girvan, M., Newman, M.: Community structure in social and biological networks. *PNAS* **99**(12) (2002)
9. Khadivi, A., Hasler, M.: A weighting scheme for enhancing community detection in networks. In: 2010 ICC. *IEEE* (2010)
10. Khadivi, A., Rad, A.A., Hasler, M.: Network community-detection enhancement by proper weighting. *Phys. Rev. E* **83**(4) (2011)
11. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**(4) (2008)
12. Leskovec, J., Krevl, A.: SNAP Datasets. <http://snap.stanford.edu/data> (2014)
13. Lyzinski, V., Tang, M., Athreya, A., Park, Y., Priebe, C.E.: Community detection and classification in hierarchical stochastic blockmodels. *IEEE Trans. Netw. Sci. Eng.* **4**(1) (2016)
14. Moradi-Jamei, B., Shakeri, H., Poggi-Corradini, P., Higgins, M.J.: A new method for quantifying network cyclic structure to improve community detection. *Stat. Phys. A* (2020)
15. Muscoloni, A., Cannistraci, C.V.: Rich-clubness test (2017). [arXiv:1704.03526](https://arxiv.org/abs/1704.03526)
16. Muscoloni, A., Cannistraci, C.V.: Leveraging the nonuniform pso network model as a benchmark for performance evaluation in community detection and link prediction. *New J. Phys.* **20**(6) (2018)
17. Muscoloni, A., Cannistraci, C.V.: A nonuniform popularity-similarity optimization (npso) model to efficiently generate realistic complex networks with communities. *New J. Phys.* **20**(5) (2018)
18. Muscoloni, A., Thomas, J.M., Ciucci, S., Bianconi, G., Cannistraci, C.V.: Machine learning meets complex networks via coalescent embedding in the hyperbolic space. *Nat. Commun.* **8**(1) (2017)
19. Orman, G.K., Labatut, V., Cherifi, H.: Comparative evaluation of community detection algorithms. *J. Stat. Mech. Theory Exp.* **2012**(08) (2012)
20. Papadopoulos, F., Kitsak, M., Serrano, M.Á., Boguná, M., Krioukov, D.: Popularity versus similarity in growing networks. *Nature* **489**(7417) (2012)
21. Peel, L., Larremore, D.B., Clauset, A.: The ground truth about metadata and community detection in networks. *Sci. Adv.* **3**(5) (2017)
22. Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: *Int. Symp. Comput. Inf. Sci.* Springer (2005)
23. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. *PNAS* **101**(9) (2004)
24. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **76**(3) (2007)
25. Ravasz, E., Barabási, A.: Hierarchical organization in complex networks. *Phys. Rev. E* **67**(2) (2003)
26. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. *PNAS* **105**(4) (2008)
27. Savkli, C., Schwartz, C., Galante, A., Cohen, J.: Novel edge and density metrics for link cohesion. In: *Int. Conf. on Comp. Netw.* Springer (2019)
28. Schaub, M.T., Delvenne, J., Rosvall, M., Lambiotte, R.: The many facets of community detection in complex networks. *Appl. Netw. Sci.* **2**(1) (2017)
29. Schwartz, C.: Analyzing semi-local link cohesion to detect communities and anomalies in complex networks. Ph.D. thesis (2021)

30. Yang, Z., Algesheimer, R., Tessone, C.J.: A comparative analysis of community detection algorithms on artificial networks. *Sci. Rep.* **6**(1) (2016)
31. Zhang, H., Zhou, C., Liang, X., Zhao, X., Li, Y.: A novel edge weighting method to enhance network community detection. *IEEE Trans. Man. Cybern. Syst.*, p. 2015 (2015)



Network Based Methodology for Characterizing Interdisciplinary Expertise in Emerging Research

Aditi Mallavarapu^{1(✉)}, Erin Walker², Cassandra Kelley², Shari Gardner³,
Jeremy Roschelle¹, and Stephen Uzzo⁴

¹ Digital Promise Global, Berkeley, CA, USA
amallavarapu@digitalpromise.org

² University of Pittsburgh, Pittsburgh, PA, USA

³ SRI International, Menlo Park, CA, USA

⁴ National Museum of Mathematics, New York, NY, USA

Abstract. Funding agencies and researchers are placing increasing emphasis on interdisciplinary research (IDR) to promote innovation and to address complex real-world problems. Understanding characteristics of IDR early (as soon as projects get funded), can formatively shape a research community at portfolio, project, and individual investigator levels. This involves surfacing the interacting components and the context that manifest IDR. We present a network-based methodology to model and analyze IDR, and apply it to a three-year portfolio of awards in the Research on Emerging Technologies for Teaching and Learning program. Our IDR analysis features two network-based metrics (1) diversity of expertise and (2) intensity of inter-expertise collaboration. It reveals the organization of the 116 expertise areas that form the “building blocks” for IDR in this community, while also highlighting potential for knowledge integration, specifically within “hotspot” topics. It also reveals gaps in IDR potential. Applying our network-based methodology for understanding IDR could enable other research domains and communities to conduct early and rapid analyses of the emerging IDR profile in their network, and could inform formative efforts to strengthen IDR.

Keywords: interdisciplinary research · scientific collaboration · network analysis · scientific expertise

1 Introduction

To enhance the productivity and efficiency of scientific research, the concept of interdisciplinarity has gained significant attention [1, 4]. For example, United States, National Science Foundation (NSF), European Union, and Swiss National Science Foundation (SNSF) have undertaken several interdisciplinary initiatives (e.g., Convergence Accelerator, Research on Emerging Technology for teaching and learning - RETTL, Horizon 2020). Emphasizing more cross-disciplinary proposals [19], these programs are specifically designed to focus on complex societal challenges aligned to national interest [13].

NSF defines interdisciplinary research (IDR) as research that “*integrates information, data, techniques, tools, perspectives, concepts or theories from two or more disciplines or bodies of specialized knowledge, performed by teams or individuals to advance the fundamental understanding or solve problems whose solutions are beyond the scope of a single discipline or area of research practice*” [20, p. 2]. Although the concept is defined and portfolios of projects are awarded, few contextual indicators of research [8] are available to analyze IDR in early years of an initiative. Existing approaches (e.g., bibliometrics, social network analysis, topic modeling, see, [23, 25, 29]) yield lagging indicators; they have often only investigated IDR for mature portfolios of research that have many publications or that require network participants to take cumbersome surveys. These techniques don’t apply in early days of an interdisciplinary initiative. The need for early indicators is important because understanding the nature of IDR early in the life cycle of an initiative could strengthen cross-disciplinary interactions, navigate gaps, connect overlapping efforts and highlight opportunities.

We contribute a practical network-based IDR methodology that can readily be applied as soon as initial awards are made in an interdisciplinary funding program. A particular novel aspect of this methodology combines the network analysis with backwards tracing from research awards to researcher biographies. We analyze expertise collaboration networks (ECNs) by tracing from award documents to PI and co-PI websites which reveal their existing expertise; through network-based metrics, we examine the potential to connect expertises in projects. To illustrate benefits of this methodology, we take a case-based [37] approach. The case comprises awards within the NSF Research on Emerging Technology for Teaching and Learning (RETTL¹) program between 2021-23. The program requests research proposals that form teams with both computer science and learning sciences expertise to explore innovative designs of technology for use in educational settings (e.g. new resources for learning in schools and museums, see Sect. 4.1 for details). Using this case of RETTL, we (1) identify and highlight varied “building blocks” of expertise, (2) identify collaborations that connect expertises, and (3) identify dense “hotspots” and rare expertise combinations. Thus we explore the potential of the ECN methodology to reveal the nature of IDR.

Our approach identifies both, what expertise is available and how the team structures that result from research awards bring or do not bring expertises together. By analogy, the behavior of a compound cannot be predicted from the behavior of the constituent elements alone, but is emergent from the conditions of reactions between elements. We capture the early potential for IDR by examining structures of planned interactions among expertises.

After a review of existing IDR methodologies and associated challenges, we describe our methodology, present the RETTL case and discuss prospective applications of ECN analysis beyond this specific case.

¹ <https://www.nsf.gov/pubs/2020/nsf20612/nsf20612.htm>.

2 Related Work

Over recent years, the escalated interest in IDR has prompted development of performance measures for IDR portfolios (for e.g., [12, 34, 35]), drawing on either science of science scholarship or team science scholarship.

Science of science studies see IDR as a phenomenon that emerges from within a larger knowledge system, which includes external drivers (e.g., complex problems, funding priorities). Studies use publication-based indicators. The methodology identifies disciplines and subdisciplines within an established hierarchy of scientific publications, and thereby how a portfolio has connected expertises. Researchers have used computational methods relating to natural language processing like topic modeling (e.g., [10, 17, 22, 23, 28]) to identify sub-disciplines within publications and outcome reports. These use statistical methods (e.g., word embeddings) to cluster the keywords that indicate facets of IDR. The identified topics are then subjected to manual expert judgment to arrange them hierarchically against backdrop of existing scientific publications. The accuracy of these methods relies on the availability of large bodies of text (e.g., publications, project reports), which pragmatically speaking, are absent for newly awarded projects.

Team science studies represent the science system as a network of people, disciplines and projects. These studies defines IDR through network-based measures like number of components, diameter, density, and transitivity of the networks (e.g., [5, 9, 11]) and complex aggregative measures like betweenness, centrality, diversity, and entropy (e.g., [15, 25, 29, 33]). This approach provides the scientific tools to evaluate the structure of collaboration for IDR within the networks [2] across multiple granularities (e.g., authors, project teams and publications). Some studies [5–9, 16] leverage advances in graph-based machine learning and visualization techniques to distill patterns of interactions and structure of the relationships among and across people and disciplines. Despite advances, there are pragmatic challenges. Data collection is difficult; response rates are low as researchers fail to respond to surveys due to their personal and time-consuming nature. Thus surveys are replaced with data from reports, publications, citations, and public policies (e.g., [12, 14]). Some researchers [27, 31] have used authors' current affiliations (e.g., discipline of highest degree, institution of affiliation) to investigate IDR within co-authorship networks. However, these indicators provide a narrow, overly formulaic window on the true range of expertise available.

Many of the above methods are lagging because they rely on output components of IDR. We focus on expertise inputs as a way of better understanding researchers' thought processes. Our methods are inspired by an intersection between a science of science approach, where we use research topics to identify overlaps and gaps in IDR, and by a team science approach, where we identify patterns of collaboration. We seek to process publicly-available data in ways that take context into account to drive understanding of newly-funded initiatives at different granularities.

3 Expertise-Collaboration Network (ECN) Methodology

3.1 Indicators

Researchers are known to include both past and current research interests in their curriculum vitae and their professional websites indicative of the depth and breadth of their work. These public artifacts describe changes that take place as careers evolve [26]. Although a specific research proposal may explicitly name just one or two expertise areas, researchers can bring their more comprehensive career experience to bear.

3.2 Expertise-Collaboration Network (ECN) Model

An ECN is an undirected graph that maps the expertise among the collaborating investigators (PI, co-PI) for a funded project. Each award, investigator and area of expertise is represented as a node, and edges connect these entities to represent collaboration. Figure 1 (left), represents a schematic of ECN for a community. The award (blue nodes) is connected to investigator (red nodes) with an “works_on” relationship and with areas of expertise (green nodes) via the “has_expertise” edge. ECN for a project represents the potential for social knowledge integration as investigators begin teamwork. Note that awards funded as collaborative initiatives (separate awards to form a cross-institutional project) are connected with an “in_collaboration” edge. We included the award nodes as opposed to single project nodes to account for the comparatively larger teams that cumulative award amounts in the collaborative endeavors afford. The network representation identifies the “building blocks” of expertise. Frequencies of expertise at the investigator, project and community level can reveal potential for IDR.

3.3 Measures

We use two network measures to analyze an ECN:

Network Diversity investigates relative differences in team configurations that may result in investigators integrating expertise for a project, (how experts share interest in an area). To quantitatively compare the project teams, we used the degree of the expertise nodes within a project team ECN to calculate the diversity and then compared differences in these scores to further characterize the typology of expertise configuration. Diversity is a property of both the type (richness) and number of items (abundance). We use the degree of the expertise nodes to track the number of expertise and types of expertise being contributed in a team through Shannon’s Diversity Index (H) [32]. This measure is often used in information theory to quantify entropy in a collection of items, given by:

$$H = - \sum p_i \log p_i \quad (1)$$

Where p_i : represents the proportion of type i of items relative to the total number of items. H is sensitive to both the evenness (different types of expertise) and

abundance (number of experts contributing that expertise) for each team [32]. The H value between 1.5 to 3.5 indicates a diverse distribution. This metric operates on the project level.

Network Intensity identifies the co-present areas of expertise which can be considered as “hotspots,” within the community with intense inter-expertise collaboration. Rare combinations of areas of expertise with low intensity of inter-expertise collaborations that are idiosyncratic to a few projects within the community are also identified as novel contributions of the community.

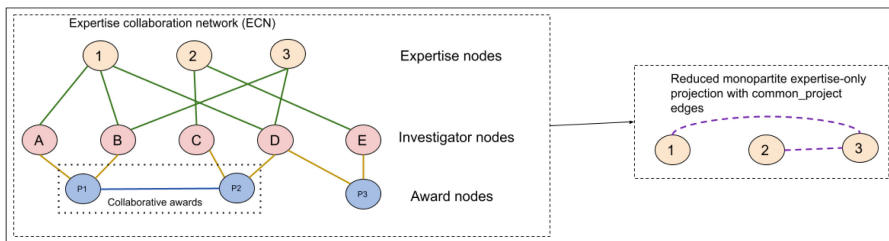


Fig. 1. Schematic projecting an expertise only monopartite graph with edges that capture the reduction process.

This metric operates at the expertise level aggregated across the projects. We measure intensity of collaboration by quantitatively reporting the frequency of repeatability and qualitatively reporting the locus of the expertise collaborations. The repeatability accounts for the intensity of co-occurrence of the expertise in the projects that are the loci of the collaborations. For this, we construct a monopartite projection of the areas of expertise (See Fig. 1 for the schematic of the process) to characterize the intensity of collaborations. The projection process reduces the ECN to only expertise nodes, preserving the co-presence on a single project with the “common_project” edges. For example, in the schematic shown in Fig. 1, the dotted purple edges between areas of expertise 1 and 3 in the projection represent co-occurring areas of expertise for the collaborative project with awards P1 and P2 (see yellow nodes) in the ECN schematic. We additionally include the number of co-presence instances as a weight of the edge, which we use to identify “hotspot” areas and rare combinations (e.g., [3]). We conceptualize the alignment between the expertise in rare combinations and the goals of the community as areas of potential growth or improvement for the community.

We used python networkX and pyvis [24] library to visualize and analyze measures for the project level sub-networks. The graph was then exported to Neo4j [21], a graph database repository that allows querying and analyzing graphs and subgraphs, which was used to create the monopartite projections. We next discuss how our approach characterized the different aspects of IDR for the RETTL community.

4 Case-Study: IDR Within RETTL Community

4.1 Research Context: RETTL Program

The NSF-funded RETTL program investigates how people learn and teach with emerging technology [18] with goals of enhancing equity, diversity, and inclusion in education [36]. The program calls for innovative ideas that draw from learning sciences and computer science to investigate educational technologies available to learners in the next 5–10 years. Simple analyses indicated the actual expertise in this research portfolio were more broad and diverse than only learning sciences and computer sciences [30], but could not reveal the nature of IDR among the many researchers and projects in the program. RETTL’s call for interdisciplinary research and its breath of actual expertise provided fertile ground for exploring the potential of our methodology to characterize IDR.

4.2 Data Collection

We collected publicly available project data from the NSF website² by searching for the identifier “Cyberlearn & Future Learn” applied to newly-awarded projects by NSF program officers. We found 87 projects and 238 investigators in awards made within 2021-23. The NSF website provided data such as a unique award number, title of the project, abstract, and names of investigators. We used the title and the abstract to identify collaborative awards (which share a single title and abstract). We deidentified the award numbers of projects and the investigator names with anonymous codes.

Data Consolidation. Terms to describe expertise had to be refined for consistency. We used an existing hierarchical tag-based framework³ that identifies popular areas of research interest in RETTL. Two researchers, who were not part of the creation of this initial framework, labelled the investigators’ expertise. The initial pass yielded low agreement 40%. We attribute the low agreement to the fact that the tags within the framework were designed for labelling projects, not people’s expertise. We thereafter refined definitions of terms and generated new codes to better fit our intention. For example, we found it necessary to separate “simulation” from “modeling.” We added additional expertise that applied to people, but not necessarily to whole projects. Overall, we added 42 terms to the framework resulting in 116 unique areas of expertise from the original 597 areas of expertise collected. These included specializations beyond broad “computer science” and “learning sciences” such as medical science, psychology, neuroscience, sociology, social science, education research, anthropology, civil engineering, manufacturing and cognitive science. Within computer sciences, there were specifics like Human-Computer Interaction (HCI), machine learning and Artificial Intelligence (AI). Within Learning Sciences, there were specifics like inquiry based learning and community partnerships.

² <https://www.nsf.gov/awardsearch/>.

³ <https://circls.org/project-tag-map>.

4.3 Analyzing Available Expertise in the RETTL Community

We constructed an ECN for each project ($N = 87$), where collaborative awards were treated as a unified initiative. In particular, the ECN network for the RETTL community formed by connecting 238 investigators, 116 areas of expertise with 112 newly funded initiatives (blue nodes), between 2021-23. The top ten specialization areas which included HCI, cognitive psychology/cognitive science, Educational technology, Machine learning, Neuroscience, AI, Augmented/virtual/mixed reality and few other expertise accounted for 32.97% of identified expertise among 73.94% experts in the community. The strong emphasis in HCI, which draws from both learning sciences and computer science research, aligns to RETTL's goal. Areas related to equity and ethics including expertise in special populations, broadening participation, ethics, bias and equity were identified to be moderately popular, but towards the lower end of the frequencies. We also investigated areas of expertise with very low frequencies. Interestingly, these were uniquely concentrated in ten RETTL projects and included at least one disciplinary expert from medical science, material science, civil engineering, industrial engineering, epidemiology, nanotechnology, philosophy, mineral processing and sociology disciplines, indicative of novel educational applications.

Investigating Expertise Configurations in Projects. The ECN revealed that investigators on projects were integrating expertise in seemingly different ways. In the examples shown below, both projects are three part collaborative projects with four investigators. In Project A (left in 2, $H = 1.22$), social welfare, development and responsibility expertise (studies conducted for socially responsive technology design), natural language processing and collaboration are shared among at least two team members, while all other 14 areas of expertise are uniquely contributed. Contrastingly, in Project B (right, in Fig. 2, $H = 0.95$), all four investigators shared expertise (AI, HCI, cognitive science, data mining, math education, intelligent tutoring systems) except for investigator 274, who contributes (right, upper left corner) four unique areas of expertise to the project (creativity, affect and emotion in learning, software engineering, education).

In addition to qualitatively documenting the expertise on each project, we also consider the different ways expertise is distributed. We identified three structural arrangements: (1) dispersed arrangement, where each expert on the team contributes unique areas of expertise having a relatively high diversity ($H > 1$), (2) shared arrangement, where investigators mostly share expertise with few investigators bringing in specialized expertise that is rare ($0.61 < H < 0.99$), and (3) collective arrangement, where investigators all share the expertise associated with the project with low diversity ($H < 0.6$). Figure 3 shows examples of different diversity categorization. Most projects had a dispersed configuration ($N = 49$). Very few projects were categorized to be collective ($N = 7$). Our future work will relate configurations to other metrics like project outcomes and publications. Note that the diversity metric is not designed to make strong claims about the extent of diversity of each team, rather we only make relative comparisons.

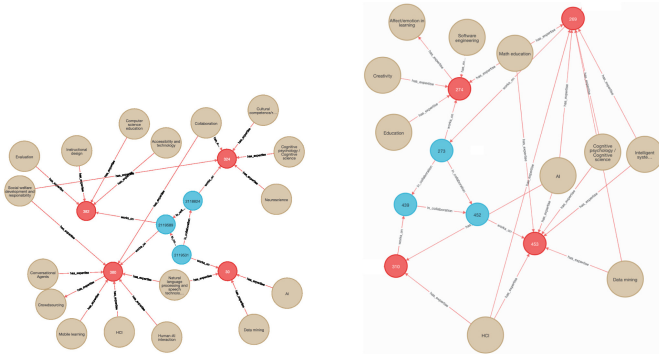


Fig. 2. ECN of Project A (left) and Project B (right) with different team configurations of expertise sharing.



Fig. 3. Three different diversity configurations - dispersed arrangement (left, $H = 1.12$), shared arrangement (center, $H = 0.91$) and collective arrangement (right, $H = 0.6$) informing diversity of expertise integration.

IDR “Hotspots” and Growth Opportunities. We also found that active expertise varied across time. This variation is indicative of the highly-emergent research undertaken by the community which is responsive to new technologies and educational possibilities [30]. We leveraged the edge weight between two expertises in the expertise-only projection of the ECN for each year to determine the high and low intensity areas in the community. In social network analysis, hotspots are places with significantly high relationship traffic relative to others. A greater frequency of related areas of expertise is indicative of a “hotspot” for the community, whereas low intensity areas can help identify opportunities for growth.

Evolving “Hotspots” of the RETTL Community. Expertise edges between HCI and AI were found in 23 (7 in 2021, 8 in 2022, 8 in 2022) project collaborations, HCI and cognitive psychology/cognitive science were found in 21 (9 in 2021, 6 in 2022, 6 in 2022) project collaborations, neuroscience and cognitive psychology/cognitive science were found in 24 projects (8 in 2021, 11 in 2022, 5 in 2023),

while Educational technology and HCI expertises were found for 20 (7 in 2021, 5 in 2022, 8 in 2023) projects. These areas were identified as “hotspots” for this community. Despite relatively high edge frequency, the edges found in ECN for only one of the three years, were not considered as “hotspots”. For example, human-AI interaction and AI, alongside AI and Data mining were common to 7 projects in the year 2023 but had lower edge frequencies for the previous years (1 in 2021, 3 in 2022). The ascent of AI to the top 3 areas of expertise has been gradual, aligning with the huge strides in mainstream AI in recent years. The strong emphasis of HCI and cognitive psychology/cognitive science across the time periods emphasizes a human-centric approach that supports learning with educational technologies. Our future work will plan for interviews with investigators to learn more about “hotspots”.

Identifying Areas of Growth. By identifying the expertise important to the community, as highlighted in the RETTL call, as the areas of low intensity (with *edge frequency* ≤ 2 through the three years) we were able to identify the areas of potential growth for the community. One such goal for the RETTL community is to combine computer science and learning sciences research to address equity. This objective is represented by six areas of expertise in our analysis: ethical AI, equity in education, ethics of technology design, bias and equity in AI, special populations, and broadening participation for underrepresented communities. These six expertise currently represent a smaller area of focus for the RETTL community, with an average of 5.66 investigators ($SD = 3.8$) and 5 projects ($SD = 3.2$) related to each expertise. The co-presence of these expertise with other expertise was observed to have low intensity (*edge frequency* ≤ 2) which dynamically changed through the years. For example, we found that ethics- and equity-related expertises were connected to other expertise with a low edge frequency indicating a small number of collaborations with experts in that area. While research that involved creating educational technology for special populations was an expertise which had a low intensity in the year 2022 and 2023, but was not included in the projects for the year 2021. Moreover, we found that often just one investigator brought these expertise to the relatively small number of projects that worked in this vein of research. These findings point to needs for improvement of how equity related expertise is integrated into the work by this community, guiding the research portfolio towards its goal.

5 Discussions and Conclusions

Our ECN methodology characterizes the emergent nature of IDR for newly funded projects. It leverages expertise of the involved researchers as an attribute of IDR, and can be extended to include more traditional indicators for IDR (e.g., surveys, publications), in turn affording a view into the knowledge integration process within the teams. Through a case-based approach, we highlighted the value of using the methodology to identify expertise, and the configurations and intensities of integration of expertise within a interdisciplinary community. This

methodology could be applied to any research community by collating the expertise of the researchers involved in the conception of the awarded projects, which is often publicly available information. Such computation of useful indicators of interdisciplinarity early in the lifecycle of the research portfolio, well before publication-based indicators are available, can help communities knowledgeably navigate urgent and innovative research to address complex real-world problems they are tasked to solve. An important aspect of this contribution is its value in understanding the expertise integration within teams and its alignment with the community goals that identified “hotspots” and opportunities for growth. We argue that such insight could act as touch points for researchers not yet involved in the community to contribute in an influential manner. This is rather contradictory to publication based IDR investigations which have stopped short of identifying areas of growth.

The utility of a network based approach is multi-granular, as it examines the qualities of a portfolio, its projects, and the participating researchers. The knowledge of different configurations of expertise could prove consequential for the projects and community. For example, at the micro level, identifying low intensity but valued areas like equity, the community can better manage human capital. At a meso level, identifying the different IDR team and expertise configurations can be useful for research communities to create resources (e.g., panel discussions and working groups to identify gaps and expose overlaps) that drive innovation. At the macro (portfolio) level, it can inform high quality peer review process by identifying the expertise available in the research community. Overall, we foresee that early analysis of IDR patterns via ECN could help funders, project leaders, and individual researchers to act more quickly and with greater insight as they forge innovative research directions and encourage improvements in interdisciplinary scholarship.

Acknowledgements. This material is based upon work supported by the National Science Foundation under Grant No. 2021159.

References

1. Allenby, B.R., Sarewitz, D.: *The techno-human condition*. MIT Press (2011)
2. Borgatti, S.P., Everett, M.G., Johnson, J.C.: *Analyzing social networks*. Sage (2018)
3. Börner, K., et al.: Mapping the co-evolution of artificial intelligence, robotics, and the internet of things over 20 years (1998–2017). *PLoS ONE* **15**(12 December), 1–21 (2020). <https://doi.org/10.1371/journal.pone.0242984>
4. Borrego, M., Newswander, L.K.: Definitions of interdisciplinary research: Toward graduate-level interdisciplinary learning outcomes. *Rev. High. Educ.* **34**(1), 61–84 (2010)
5. Calero Valdez, A., et al.: Using mixed node publication network graphs for analyzing success in interdisciplinary teams. In: Huang, R., Ghorbani, A.A., Pasi, G., Yamaguchi, T., Yen, N.Y., Jin, B. (eds.) *AMT 2012. LNCS*, vol. 7669, pp. 606–617. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35236-2_61

6. Cunningham, E., Greene, D.: The Structure of Interdisciplinary Science: Uncovering and Explaining Roles in Citation Graphs. *Studies in Computational Intelligence* 1077 SCI, 364–376 (2023). https://doi.org/10.1007/978-3-031-21127-0_30
7. Dossou Kpanou, B., Kelsey, K., Bower, K.: An evaluation of social networks within federally funded research projects. *Adv. Agric. Dev.* **1**(3), 42–54 (2020). <https://doi.org/10.37433/aad.v1i3.65>
8. Dörpinghaus, J., Klante, S., Christian, M., Meigen, C., Düing, C.: From social networks to knowledge graphs: a plea for interdisciplinary approaches. *Soc. Sci. Humanities Open* **6**(1), 100,337 (2022). <https://doi.org/10.1016/j.ssaho.2022.100337>. <https://www.sciencedirect.com/science/article/pii/S2590291122000912>
9. Ghani, S., Kwon, B.C., Lee, S., Yi, J.S., Elmqvist, N.: Visual analytics for multimodal social network analysis: a design study with social sixientists. *IEEE Trans. Visual Comput. Graphics* **19**(12), 2032–2041 (2013). <https://doi.org/10.1109/TVCG.2013.223>
10. Gomez, M.J., Ruiperez-Valiente, J.A., Garcia Clemente, F.J.: Analyzing trends and patterns across the educational technology communities using Fontana framework. *IEEE Access* **10**, 35336–35351 (2022). <https://doi.org/10.1109/ACCESS.2022.3163253>
11. Hicks, D.J., Coil, D.A., Stahmer, C.G., Eisen, J.A.: Network analysis to evaluate the impact of research funding on research community consolidation. *PLoS ONE* **14**(6), 1–14 (2019). <https://doi.org/10.1371/journal.pone.0218273>
12. Klein, J.T.: Evaluation of interdisciplinary and transdisciplinary research: a literature review. *Am. J. Prev. Med.* **35**(2), S116–S123 (2008)
13. Klein, J.T.: A conceptual vocabulary of interdisciplinary science. In: *The Oxford Handbook of Interdisciplinarity*, pp. 3–14. Oxford University Press (2010)
14. Kohli, M., Jain, S.: Analysis of co-authorship network of scientists working on topic of network theory. *Int. J. Digital Appl. Contemporary Res.* **6**(08), 1–8 (2018)
15. Leydesdorff, L.: Betweenness centrality as an indicator of the interdisciplinarity of scientific journals. *J. Am. Soc. Inform. Sci. Technol.* **58**(9), 1303–1319 (2007)
16. Mall, R., Langone, R., Suykens, J.A.: FURS: Fast and Unique Representative Subset selection retaining large-scale community structure. *Soc. Netw. Anal. Min.* **3**(4), 1075–1095 (2013). <https://doi.org/10.1007/s13278-013-0144-6>
17. Meyers, A., He, Y., Glass, Z., Babko-Malaya, O.: The termolator: terminology recognition based on chunking, statistical and search-based scores. *CEUR Workshop Proceedings* **1384**(January), 34–43 (2015). <https://doi.org/10.3389/frma.2018.00019>
18. Motz, B.A., et al.: A LAK of direction: misalignment between the goals of learning analytics and its research scholarship. *J. Learn. Anal.* 7750, 1–13 (2023). <https://learning-analytics.info/index.php/JLA/article/view/7913>
19. National Academies of Sciences Engineering Medicine & others: *Fostering the culture of convergence in research: proceedings of a workshop*. National Academies Press (2019)
20. National Academy of Sciences & National Academy of Engineering & Institute of Medicine: *Facilitating Interdisciplinary Research*. The National Academies Press, Washington, DC (2005). <https://doi.org/10.17226/11153>. <https://nap.nationalacademies.org/catalog/11153/facilitating-interdisciplinary-research>
21. Neo4j: Neo4j - the world's leading graph database (2012). <http://neo4j.org/>
22. Newman, D., Bonilla, E.V., Buntine, W.: Improving topic coherence with regularized topic models. *Advances in neural information processing systems* 24 (2011)

23. Nichols, L.G.: A topic model approach to measuring interdisciplinarity at the National Science Foundation. *Scientometrics* **100**(3), 741–754 (2014). <https://doi.org/10.1007/s11192-014-1319-2>
24. Perrone, G., Unpingco, J., Lu, H.m.: Network visualizations with Pyvis and VisJS. Proceedings of the 19th Python in Science Conference, pp. 58–62 (2020). <https://doi.org/10.25080/majora-342d178e-008>
25. Porter, A., Rafols, I.: Is science becoming more interdisciplinary? measuring and mapping six research fields over time. *Scientometrics* **81**(3), 719–745 (2009)
26. Porter, A.L., Cohen, A.S., David Roessner, J., Perreault, M.: Measuring researcher interdisciplinarity. *Scientometrics* **72**(1), 117–147 (2007). <https://doi.org/10.1007/s11192-007-1700-5>
27. Porter, A.L., Roessner, D.J., Heberger, A.E.: How interdisciplinary is a given body of research? *Res. Eval.* **17**(4), 273–282 (2008)
28. Rafols, I., Leydesdorff, L.: Content-based and algorithmic classifications of journals: Perspectives on the dynamics of scientific communication and indexer effects. *J. Am. Soc. Inform. Sci. Technol.* **60**(9), 1823–1835 (2009)
29. Rafols, I., Meyer, M.: Diversity and network coherence as indicators of interdisciplinarity: case studies in bionanoscience. *Scientometrics* **82**(2), 263–287 (2010)
30. Roschelle, J., et al.: AI and the Future of Learning: Expert Panel Report Suggested Citation Acknowledgements. Tech. Rep. November (2020). <https://circls.org/reports/ai-report>
31. Schummer, J.: Interdisciplinary issues in nanoscale research. Heidelberg (2004)
32. Spellerberg, I.F., Fedor, P.J.: A tribute to claude shannon (1916–2001) and a plea for more rigorous use of species richness, species diversity and the ‘shannon-wiener’ index. *Global Ecol. Biogeography* **12**(3), 177–179. [https://doi.org/10.1046/j.1466-822X.2003.00,015.x\(2003\)](https://doi.org/10.1046/j.1466-822X.2003.00,015.x(2003))
33. Stirling, A.: A general framework for analysing diversity in science, technology and society. *J. R. Soc. Interface* **4**(15), 707–719 (2007)
34. Stokols, D., Hall, K.L., Taylor, B.K., Moser, R.P.: The science of team science: overview of the field and introduction to the supplement. *Am. J. Prev. Med.* **35**(2), S77–S89 (2008)
35. Stokols, D., Misra, S., Moser, R.P., Hall, K.L., Taylor, B.K.: The ecology of team science: understanding contextual influences on transdisciplinary collaboration. *Am. J. Prev. Med.* **35**(2), S96–S115 (2008)
36. White House: Fact Sheet: Biden-Harris Administration Secures Voluntary Commitments from Leading Artificial Intelligence Companies to Manage the Risks Posed by AI. <https://www.whitehouse.gov/wp-content/uploads/2023/07/Ensuring-Safe-Secure-and-Trustworthy-AI.pdf>
37. Yin, R.K.: Case study research and applications. Sage (2018)



Classification Supported by Community-Aware Node Features

Bogumił Kamiński¹(✉), Paweł Prałat², François Théberge³,
and Sebastian Zając¹

¹ Decision Analysis and Support Unit, SGH Warsaw School of Economics, Warsaw,
Poland

{bkamins,szajac2}@sgh.waw.pl

² Department of Mathematics, Toronto Metropolitan University, Toronto, ON,
Canada

pralat@torontomu.ca

³ Tutte Institute for Mathematics and Computing, Ottawa, ON, Canada
theberge@ieee.org

Abstract. A community structure that is often present in complex networks plays an important role not only in their formation but also shapes dynamics of these networks, affecting properties of their nodes. In this paper, we propose a family of community-aware node features and then investigate their properties. We show that they have high predictive power for classification tasks. We also verify that they contain information that cannot be recovered completely neither by classical node features nor by classical or structural node embeddings.

Keywords: node classification · community structure · node features

1 Introduction

In the context of relational data, node classification is a particularly important problem in which data is represented as a network and the goal is to predict labels associated with its nodes. It is widely used in various practical applications such as recommender systems [27], social network analysis [3], and applied chemistry [12].

However, for classifiers to perform well, they must have access to a set of highly informative node features that can discriminate representatives of different classes. No matter how sophisticated classifiers one builds, they will perform poorly as long as they do not get informative input concerning the problem at hand. Hence, it is desirable to enrich a family of available features and apply machine learning tools to features of various sorts.

In this paper, we investigate a family of features that depend on the community structure that is often present in complex networks and plays an important role in their formation, affecting nodes' properties. Such features are further called *community-aware features*. Indeed, community structure of real-world

networks often reveals the internal organization of nodes [10]. Such communities form groups of densely connected nodes with substantially less edges touching other parts of the graph. Identifying communities in a network can be done in an unsupervised way and is often the first step the analysts take.

The motivation to study community-aware features is twofold. On one hand, one can expect that for many node classification tasks such features can be highly informative. For example, it might be important whether a given node is a strong community member or, conversely, it is loosely tied to many communities. On the other hand, one can expect that community-aware features are not highly correlated to other features that are typically computed for networks. Indeed, to compute community-aware features one needs first to identify the community structure of a graph. This, in turn, is a complicated non-linear transformation of the input graph, which cannot be expected to be easily recovered by supervised or unsupervised machine learning models that are not designed to be community-aware.

In this paper, we show that there are classes of node prediction problems in which community-aware features have high predictive power. We also verify that community-aware features contain information that cannot be recovered completely neither by classical node features nor by node embeddings (both classical as well as structural). In our experiments, we concentrate on binary classification to assure that the results can be reported consistently across different graphs.

There are some community-aware features already introduced in the literature such as CADA [15] (Community Aware Detection of Anomalies) or the participation coefficient [14]; see Sect. 2 for their definitions. However, it is important to highlight that both CADA and the participation coefficient ignore the distribution of community sizes. We argue that taking community sizes into account when computing community-aware features matters as it provides a more detailed picture. Therefore, in this paper we propose a class of community-aware features that, via the appropriate null model, take into account community sizes and compare their predictive performance to the measures that have been previously proposed in the literature.

This is a short proceeding version of a longer paper¹. The longer version includes a discussion about using the null models to design ML tools, including the modularity function that is used by many clustering algorithms. More importantly, in the longer version we show how the null model is used to define one of our community-aware features, namely, the Community Association Strength (see Subsect. 2.3). Due to space limitations, we also do not include in this short version various additional experiments (for example, on synthetic networks) and discussion on computational complexity of the algorithms computing node features.

¹ The preprint of the longer version can be found on-line: <https://math.torontomu.ca/~pralat/research.html>.

2 Community-Aware Node Features

In this section, we introduce various community-aware node features. All of them aim to capture and quantify how given nodes are attached to communities. It will be assumed that a partition $\mathbf{A} = \{A_1, A_2, \dots, A_\ell\}$ of the set of nodes V into ℓ communities is already provided; communities induced by parts A_i ($i \in [\ell]$) are denser comparing to the global density of the graph. Such partition can be found by any clustering algorithm. In our empirical experiments we use Leiden [25] which is known to produce good, stable results.

To simplify the notation, we will use $\deg_{A_i}(v)$ to be the number of neighbours of v in A_i .

2.1 Anomaly Score CADA

The first community-aware node feature is the anomaly score introduced in [16]. The *anomaly score* is computed as follows: for any node $v \in V$ with $\deg(v) \geq 1$,

$$\text{cd}(v) = \frac{\deg(v)}{d_{\mathbf{A}}(v)}, \quad \text{where} \quad d_{\mathbf{A}}(v) = \max \left\{ \deg_{A_i}(v) : A_i \in \mathbf{A} \right\};$$

the denominator, $d_{\mathbf{A}}(v)$, represents the maximum number of neighbouring nodes that belong to the same community. In one extreme, if all neighbours of v belong to the same community, then $\text{cd}(v) = 1$. In the other extreme, if no two neighbours of v belong to the same community, then $\text{cd}(v) = \deg(v)$.

Note that $\text{cd}(v)$ does not pay attention to which community node v belongs to. Moreover, this node feature is unbounded, that is, $\text{cd}(v)$ may get arbitrarily large. As a result, we will also investigate the following small modification of the original score, the *normalized anomaly score*: for any node $v \in A_i$ with $\deg(v) \geq 1$,

$$\overline{\text{cd}}(v) = \frac{\deg_{A_i}(v)}{\deg(v)}.$$

Clearly, $0 \leq \overline{\text{cd}}(v) \leq 1$. Moreover, any reasonable clustering algorithm typically should try to assign v to the community where most of its neighbours are, so most nodes are expected to have $\overline{\text{cd}}(v) = 1/\text{cd}(v)$.

2.2 Normalized Within-Module Degree and Participation Coefficient

In [14], an interesting and powerful approach was proposed to quantify the role played by each node within a network that exhibits community structure. Seven different universal roles were heuristically identified, each defined by a different region in the $(z(v), p(v))$ 2-dimensional parameter space, where $z(v)$ is the normalized within-module degree of a node v and $p(v)$ is the participation coefficient of v . Node feature $z(v)$ captures how strongly a particular node is connected to

other nodes within its own community, completely ignoring edges between communities. On the other hand, node feature $p(v)$ captures how neighbours of v are distributed between all parts of the partition \mathbf{A} .

Formally, the *normalized within-module degree* of a node v is defined as follows: for any node $v \in A_i$,

$$z(v) = \frac{\deg_{A_i}(v) - \mu(v)}{\sigma(v)},$$

where $\mu(v)$ and $\sigma(v)$ are, respectively, the mean and the standard deviation of $\deg_{A_i}(u)$ over all nodes u in the part v belongs to. If node v is tightly connected to other nodes within the community, then $z(v)$ is large and positive. On the other hand, $|z(v)|$ is large and $z(v)$ is negative when v is loosely connected to other peers.

The *participation coefficient* of a node v is defined as follows: for any node $v \in V$ with $\deg(v) \geq 1$,

$$p(v) = 1 - \sum_{i=1}^{\ell} \left(\frac{\deg_{A_i}(v)}{\deg(v)} \right)^2.$$

The participation coefficient $p(v)$ is equal to zero if v has neighbours exclusively in one part (most likely in its own community). In the other extreme situation, the neighbours of v are homogeneously distributed among all parts and so $p(v)$ is close to the trivial upper bound of $1 - 1/\ell \approx 1$.

2.3 Community Association Strength

As already advertised, let us now introduce our own community-aware node feature that takes the distribution of community sizes into account. Its derivation is explained in the longer version of this paper. For any $v \in A_i$, we define the *community association strength* as follows:

$$\beta^*(v) = 2 \left(\frac{\deg_{A_i}(v)}{\deg(v)} - \lambda \frac{\text{vol}(A_i) - \deg(v)}{\text{vol}(V)} \right).$$

The lower the value of $\beta^*(v)$, the less associated node v with its own community is. In the derivation above we allow for any $\lambda > 0$, but in the experiments, we will use $\lambda = 1$.

Let us also notice that when $\lambda = 1$, $\beta^*(v)$ is essentially twice the normalized anomaly score $\text{cd}(v)$ after adjusting it to take into account the corresponding prediction from the null model. A simplified version of this node feature was already used in [18].

2.4 Distribution-Based Measures

Our next community-aware node features are similar in spirit to the participation coefficient, that is, they aim to measure how neighbours of a node v are

distributed between all parts of the partition \mathbf{A} . The main difference is that they pay attention to the sizes of parts of \mathbf{A} and compare the distribution of neighbours to the corresponding predictions from the null model. They are upgraded versions of the participation coefficient, similarly to the community association strength being an upgraded counterpart of the normalized anomaly score.

Formally, for any node $v \in V$, let $q_1(v)$ be the vector representing fractions of neighbours of v in various parts of partition \mathbf{A} . Similarly, let $\hat{q}_1(v)$ be the corresponding prediction for the same vector based on the Chung-Lu model. Note that $\hat{q}_1(v) = \hat{q}_1$ does *not* depend on v (of course, it should not!) but only on the distribution of community sizes. Our goal is to measure how similar the two vectors are. A natural choice would be any p -norm, but since both vectors are stochastic (that is, all entries are non-negative and they add up to one), one can also use any good measure for comparison of probability distributions. In our experiments we tested the following node features: L^1 norm $L_1^1(v)$, L^2 norm $L_1^2(v)$, *Kullback-Leibler divergence* $\text{kl}_1(v)$, and *Hellinger distance* $h_1(v)$.

The above measures pay attention to which communities neighbours of v belong to. However, some of such neighbours might be strong members of their own communities but some of them might not be. Should we pay attention that? Is having a few strong members of community A_i as neighbours equivalent to having many neighbours that are weak members of A_i ? To capture these nuances, one needs to consider larger ego-nets around v , nodes at distance at most 2 from v . We define $q_2(v)$ to be the average value of $q_1(u)$ taken over all neighbours of v . As before, $\hat{q}_2(v)$ is the corresponding prediction based on the null model. However, since $\hat{q}_1(u) = \hat{q}_1$ does not depend on u , $\hat{q}_2(v)$ also does not depend on v and, in fact, it is equal to \hat{q}_1 . The difference between $q_2(v)$ and $\hat{q}_2(v)$ may be measured by any metric used before. In our experiments we tested $L_2^1(v)$, $L_2^2(v)$, $\text{kl}_2(v)$, and $h_2(v)$, counterparts of $L_1^1(v)$, $L_1^2(v)$, $\text{kl}_1(v)$, and $h_1(v)$ respectively.

Let us mention that $q_1(v)$ and $q_2(v)$ have a natural and useful interpretation. Consider a random walk that starts at a given node v . The i th entry of the $q_1(v)$ vector is the probability that a random walk visits a node from community A_i after one step. Vector $q_2(v)$ has the same interpretation but after two steps are taken by the random walk.

One can repeat the same argument and define $L_i^1(v)$, etc., for any natural number i by performing i steps of a random walk. Moreover, a natural alternative approach would be to consider all possible walk lengths where connections made with distant neighbours are penalized by an attenuation factor α , as it is done in the classical Katz centrality [19].

Finally, let us note that the above aggregation processes could be viewed as simplified versions of GNNs classifiers. Therefore, the investigation of these measures additionally shows how useful community-aware measures could be when used in combination with GNN models.

3 Experiments

3.1 Graphs Used

We consider undirected, connected, and simple graphs so that all node features are well defined and all methods that we use work properly. In each graph, we have some “ground-truth” labels for the nodes which are used to benchmark classification algorithms. For consistency of the reported metrics, we consider binary classification tasks, so the ground-truth node features that are to be predicted will always consist of labels from the set $\{0, 1\}$ with label 1 being the target class.

In the experiments, we used two families of graphs. The first family consists of synthetic networks, the **Artificial Benchmark for Community Detection with Outliers (ABCD+o)** [18]. The main goal of experiments on this family is to perform a sanity test to evaluate whether the basic functionality of community-aware node features is working correctly or not. The results (that are available in the longer version of this paper) show that for this class of graphs community-aware node features significantly outperform other features.

The second family of networks we used in our experiments are empirical real-world graphs. We tried to select a collection of graphs with different properties (density, community structure, degree distribution, clustering coefficient, etc.). More importantly, some of them have highly unbalanced binary classes. Experiments with these networks will serve as a more challenging and robust test for usefulness of the proposed community-aware node features.

Empirical Graphs. For experiments on real-world, empirical networks, we selected the following 5 datasets. In cases when multiple connected components were present, we kept only the giant component. Self-loops, if present, were also dropped before performing the experiments. We summarize some statistics for those graphs in Table 1.

- **Reddit** [20]: A bipartite graph with 9,998 nodes representing users in one part and 982 nodes representing subreddits in the other one. The target class represents banned users.
- **Grid** [21]: A power grid network with attributed nodes. The target class corresponds to nodes with “plant” attribute.
- **Facebook** [23]: Nodes correspond to official Facebook pages that belong to one of the 4 categories and edges are mutual likes. The target class corresponds to “politician” category.
- **LastFM** [24]: Nodes are users of the social network and edges represent mutual followers. There are some nodes attributes including the location; the target class corresponds to nodes with “country 17” attribute.
- **Amazon** [8]: Nodes are users and edges represent common product reviews. The target class corresponds to users with less than 20% “helpful” votes, and non-target correspond to users with more than 80% “helpful” votes.

Table 1. Statistics of the selected real-world empirical graphs.

dataset	# of nodes	average degree	# of clusters	target class proportion
Reddit	10,980	14.30	12	3.661%
Grid	13,478	2.51	78	0.861%
LastFM	7,624	7.29	28	20.619%
Facebook	22,470	15.20	58	25.670%
Amazon	9,314	37.49	39	8.601%

3.2 Node Features Investigated

The community-aware node features that we tested are summarized in Table 2. The features are computed with reference to a partition of a graph into communities obtained using the Leiden algorithm. The partition is chosen as the best of 1,000 independent runs of the `community_leiden` function implemented in the *igraph* library [7] (Python interface of the library was used). Each of such independent runs was performed until a stable iteration was reached.

Table 2. Community-aware node features used in our experiments. A combination of WMD and CPC is also used as a 2-dimensional embedding of a graph (WMD+CPC).

abbreviation	symbol	name	subsection
CADA	$cd(v)$	anomaly score CADA	2.1
CADA*	$\overline{cd}(v)$	normalized anomaly score	2.1
WMD	$z(v)$	normalized within-module degree	2.2
CPC	$p(v)$	participation coefficient	2.2
CAS	$\beta^*(v)$	community association strength	2.3
CD_L11	$L_1^1(v)$	L^1 norm for the 1st neighbourhood	2.4
CD_L21	$L_1^2(v)$	L^2 norm for the 1st neighbourhood	2.4
CD_KL1	$kl_1(v)$	Kullback–Leibler divergence for the 1st neighbourhood	2.4
CD_HD1	$h_1(v)$	Hellinger distance for the 1st neighbourhood	2.4
CD_L12	$L_2^1(v)$	L^1 norm for the 2nd neighbourhood	2.4
CD_L22	$L_2^2(v)$	L^2 norm for the 2nd neighbourhood	2.4
CD_KL2	$kl_2(v)$	Kullback–Leibler divergence for the 2nd neighbourhood	2.4
CD_HD2	$h_2(v)$	Hellinger distance for the 2nd neighbourhood	2.4

Classical (non-community-aware) node features are summarized in Table 3. These are standard and well-known node features. We omit their definitions but, instead, refer to the appropriate sources in the table or [17].

Finally, we will use two more sophisticated and powerful node features obtained through graph embeddings. Embeddings can be categorized into two main types: classical embeddings and structural embeddings. Classical embeddings focus on learning both local and global proximity of nodes, while structural

embeddings learn information specifically about the local structure of nodes’ neighbourhood. We test one embedding from each class: `node2vec` [13] and `struc2vec` [22].

Table 3. Classical (non-community-aware) node features that are used in our experiments.

abbreviation	name	reference
<code>lcc</code>	local clustering coefficient	[26]
<code>bc</code>	betweenness centrality	[11]
<code>cc</code>	closeness centrality	[2]
<code>dc</code>	degree centrality	[17]
<code>ndc</code>	average degree centrality of neighbours	[1]
<code>ec</code>	eigenvector centrality	[4]
<code>eccen</code>	node eccentricity	[6]
<code>core</code>	node coreness	[17]
<code>n2v</code>	16-dimensional <code>node2vec</code> embedding	[13]
<code>s2v</code>	16-dimensional <code>struc2vec</code> embedding	[22]

3.3 Experiments

In this section, we present the results of two numerical experiments that were performed to investigate the usefulness of community-aware features:

1. *information overlap* between community-aware and classical features;
2. *combined variable importance for prediction* of community-aware and classical features.

From the computational perspective, all analytical steps (generation of graphs, extractions of both community-aware and classical features, execution of experiments) were implemented in such a way that all experiments are fully reproducible. In particular, all steps that involve pseudo-random numbers were appropriately seeded. The source code allowing for reproduction of all results is available at GitHub repository².

Information Overlap. In the first experiment, our goal was to test, using a variety of models, to what extent each community-aware feature described in Table 2 can be explained by all the classical features from Table 3. For each community-aware feature, we independently measured how well it is explained by each model via computing the Kendall correlation of the value of the selected

² <https://github.com/sebkaz/BetaStar.git>.

feature and its prediction. To consider possible non-linear relationships, the non-parametric Kendall correlation was used; it checks how well the ordering of predictions matches the ordering of the target. We also used measures such as R^2 , which assumes linearity and homoskedasticity of prediction error of the relationship, and the results were similar.

The tests were performed using 70/30 train-test split of data. To ensure that the reported results are robust, for each community-aware feature, five models were built using respectively random forest, xgboost, lightgbm, linear regression and regularized regression. The maximum Kendall correlations we obtained are reported.

The goal of this experiment is to show that community-aware features cannot be explained completely by classical features (including two highly expressible embeddings). The conclusion is that it is worth to include such features in predictive models as they could potentially improve their predictive power. However, this additional information could be simply noise and so not useful in practice. To verify the usefulness of the community-aware features, we performed two more experiments, namely, *one-way predictive power* and *combined variable importance for prediction* checks. In these experiments, we check if community-aware features are indeed useful in node label prediction problems. In the longer version of this paper the results of both experiments are reported. Here, due to space limitation, we only describe the results of *combined variable importance for prediction* experiment.

In general, for empirical graphs described in Sect. 3.1, the target is a binary feature that measures some practical feature or a role of a given node. It is important to highlight that these features are not derived from the community structure of these graphs, at least not directly. Instead, they are characteristics of nodes defined independently of the graph structure. Therefore, for these networks, we do not expect that community-aware features will significantly outperform other features. However, we conjecture that in many empirical networks, it may be the case that the prediction target is related to the fact that a node is a strong member of its own community or not. We expect to see that some community-aware features are still useful in prediction. It is important to highlight that, as we have described in Sect. 3.1, we have not hand-picked a few empirical networks that present good performance of community-aware features, aiming for a diverse collection of networks.

Results and Observations. For empirical graphs, in Table 4 we observe correlations significantly bounded away from 1. (For synthetic networks the correlations are even lower—see the longer version of this paper). In particular, for the **Grid** graph, the correlation values are the lowest in the family of the empirical graphs (slightly above 0.2 for single-community measures).

In summary, the presented results confirm that the information encapsulated in community-aware measures cannot be recovered completely using classical features (even including embeddings). In the following experiments, we investigate if this extra information is useful for the node classification task.

Combined Variable Importance for Prediction. The second experiment (*combined variable importance for prediction*) provides a way to verify the usefulness of community-aware features for node classification task. For each graph we build a single model predicting the target variable (defined in the Empirical Graphs section) that takes into account all community-aware as well as all classical features (including both embeddings) as explanatory variables. A random forest classifier was built and the permutation variable importance [5, 9] measure was computed for each feature using *APS* (average precision score) as a target predictive measure.

As in the previous experiment, a 70/30 train-test split was used. We report the ranking of variable importance (rank 1 being the most important one) so that the values are comparable across all graphs investigated in this experiment. The raw importance scores have different ranges for various graphs.

Table 4. Information overlap between community-aware and classical features. The maximum of Kendall correlation between target and predictions on test data set.

target	Amazon	Facebook	Grid	LastFM	Reddit
CADA	0.5830	0.5666	0.2156	0.4815	0.6826
CADA*	0.6058	0.5828	0.2174	0.5058	0.6867
CPC	0.6338	0.5992	0.2193	0.5175	0.7193
CAS	0.6538	0.6257	0.2999	0.5594	0.7306
CD_L21	0.7052	0.6464	0.3496	0.5698	0.7574
CD_L22	0.7554	0.7355	0.3557	0.6295	0.7941
CD_L11	0.7251	0.7041	0.6978	0.6220	0.7735
CD_L12	0.7794	0.7785	0.6447	0.6884	0.7810
CD_KL1	0.7176	0.7516	0.7394	0.6289	0.7755
CD_HD1	0.7383	0.7482	0.7168	0.6459	0.7853
CD_KL2	0.7706	0.7826	0.7292	0.6853	0.8097
CD_HD2	0.8212	0.8173	0.6930	0.7369	0.8221
WMD	0.8447	0.8456	0.8488	0.8531	0.7638

Results and Observations. The results for empirical graphs are presented in Table 5. The ranks range between 1 and 53 (with rank 1 being the best), since there are 53 features in total (13 community-aware, 8 classical, 16 for `node2vec`, and 16 for `struc2vec`). The rows are sorted by the arithmetic mean of rank correlations across all graphs.

For one empirical graph (namely, the *Facebook* graph), no community-aware measure appears in the top-10. It should be noted though, as can be seen by the experiments included in the longer version of this paper, that both `node2vec` and `struc2vec` embeddings provide almost perfect prediction for this graph. On

Table 5. Variable importance ranks for community-aware features in models including all features as explanatory variables. Values range from 1 (the best) to 53 (the worst). Rows are sorted by best rank across the considered graphs.

variable	Amazon	Facebook	Grid	LastFM	Reddit
WMD	49	25	1	8	31
CD_L22	25	28	3	11	49
CD_L12	24	53	20	7	4
CAS	16	17	6	6	40
CD_HD1	23	23	17	27	8
CD_KL1	18	20	14	9	30
CD_HD2	9	52	46	38	32
CD_L21	19	32	11	29	25
CADA*	37	34	26	14	50
CD_L11	14	22	18	45	27
CD_KL2	15	31	27	42	28
CADA	26	33	22	15	33
CPC	39	30	24	17	26

the other hand, for the *Grid* graph, community-aware features are important (3 of them are in the top-10). In general, the community-aware features that score high for at least one graph are: CAS, CD_L22, WMD, CD_L12, CD_HD2, CD_HD1, and CD_KL1. In particular, we see that the second-neighbourhood measures are well represented. This indicates that looking at the community structure of larger ego-nets of nodes is useful for empirical graphs. This is not the case for synthetic **ABCD+o** graphs as their generation structure is simpler than the more sophisticated mechanisms that lead to network formation of empirical complex networks. (As before, we refer the reader to the longer version of this paper for experiments on synthetic networks.)

Acknowledgements. BK and SZ have been supported by the Polish National Agency for Academic Exchange under the Strategic Partnerships programme, grant number BPI/PST/2021/1/00069/U/00001.

References

1. Barrat, A., Barthélemy, M., Pastor-Satorras, R., Vespignani, A.: The architecture of complex weighted networks. *Proc. Nat. Acad. Sci.* **101**(11), 3747–3752 (2004)
2. Bavelas, A.: Communication patterns in task-oriented groups. *J. Acoust. Soc. Am.* **22**(6), 725–730 (1950)
3. Bhagat, S., Cormode, G., Muthukrishnan, S.: Node classification in social networks. In: Aggarwal, C. (eds.) *Social Network Data Analytics*. Springer, Boston, MA. (2011). https://doi.org/10.1007/978-1-4419-8462-3_5

4. Bonacich, P., Lloyd, P.: Eigenvector-like measures of centrality for asymmetric relations. *Soc. Netw.* **23**, 191–201 (2001)
5. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
6. Buckley, F., Harary, F.: *Distance in Graphs*. Addison-Wesley, vol. 2 (1990)
7. Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal, Complex Syst.* **1695**, 1–9 (2006). <https://igraph.org>
8. Dou, Y., Liu, Z., Sun, L., Deng, Y., Peng, H., Yu, P.S.: Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In: *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM 2020)* (2020)
9. Fisher, A., Rudin, C., Dominici, F.: All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously (2019). <https://arxiv.org/abs/1801.01489>
10. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3–5), 75–174 (2010)
11. Freeman, L.C.: A set of measures of centrality based on betweenness. *Sociometry* **40**(1), 35–41 (1977). <http://www.jstor.org/stable/3033543>
12. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: *International Conference on Machine Learning*, pp. 1263–1272. PMLR (2017)
13. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. *CoRR*, abs/1607.00653 (2016). <http://arxiv.org/abs/1607.00653>, arXiv:1607.00653
14. Guimera, R., Amaral, L.A.N.: Functional cartography of complex metabolic networks. *Nature* **433**(7028), 895–900 (2005)
15. Helling, T.J., Scholtes, J.C., Takes, F.W.: A community-aware approach for identifying node anomalies in complex networks. In: *International Workshop on Complex Networks & Their Applications* (2018)
16. Helling, T.J., Scholtes, J.C., Takes, F.W.: A community-aware approach for identifying node anomalies in complex networks. In: Aiello, L.M., Cherifi, C., Cherifi, H., Lambiotte, R., Lió, P., Rocha, L.M. (eds.) *COMPLEX NETWORKS 2018*. *SCI*, vol. 812, pp. 244–255. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05411-3_20
17. Kamiński, B., Prałat, P., Théberge, F.: *Mining Complex Networks*. Chapman and Hall/CRC (2021)
18. Kamiński, B., Prałat, P., Théberge, F.: Artificial benchmark for community detection with outliers (ABCD+o). *Appl. Netw. Sci.* **8**(1), 25 (2023)
19. Katz, L.: A new status index derived from sociometric analysis. *Psychometrika* **18**(1), 39–43 (1953)
20. Kumar, S., Zhang, X., Leskovec, J.: Predicting dynamic embedding trajectory in temporal interaction networks. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1269–1278 (2019)
21. Matke, C., Medjroubi, W., Kleinhans, D.: SciGRID - An Open Source Reference Model for the European Transmission Network (v0.2) (2016). <http://www.scigrid.de>
22. Ribeiro, L.F.R., Saverese, P.H.P., Figueiredo, D.R.: Struc2vec: learning node representations from structural identity. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 385–394 (2017). <https://doi.org/10.1145/3097983.3098061>
23. Rozemberczki, B., Allen, C., Sarkar, R.: Multi-scale attributed node embedding. *J. Complex Netw.* **9**(2), cnab014 (2021)

24. Rozemberczki, B., Sarkar, R.: Characteristic functions on graphs: birds of a feather, from statistical descriptors to parametric models. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM 2020), pp. 1325–1334. ACM (2020)
25. Traag, V.A., Waltman, L., Van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**(1), 5233 (2019)
26. Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. *Nature* **393**, 440–442 (1988)
27. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 974–983 (2018)



Signature-Based Community Detection for Time Series

Marco Gregnanin^{1,2(✉)}, Johannes De Smedt², Giorgio Gnecco¹,
and Maurizio Parton³

¹ IMT School for Advanced Studies Lucca, Lucca, Italy

² KU Leuven, Leuven, Belgium

marco.gregnanin@imtlucca.it

³ University of Chieti-Pescara, Pescara, Italy

Abstract. Community detection for time series without prior knowledge poses an open challenge within complex networks theory. Traditional approaches begin by assessing time series correlations and maximizing modularity under diverse null models. These methods suffer from assuming temporal stationarity and are influenced by the granularity of observation intervals.

In this study, we propose an approach based on the signature matrix, a concept from path theory for studying stochastic processes. By employing a signature-derived similarity measure, our method overcomes drawbacks of traditional correlation-based techniques.

Through a series of numerical experiments, we demonstrate that our method consistently yields higher modularity compared to baseline models, when tested on the Standard and Poor's 500 dataset. Moreover, our approach showcases enhanced stability in modularity when the length of the underlying time series is manipulated.

This research contributes to the field of community detection by introducing a signature-based similarity measure, offering an alternative to conventional correlation matrices.

Keywords: Community Detection · Signature · Time Series

1 Introduction

In recent years, the exploration of structural properties within complex systems has garnered considerable significance across diverse scientific domains, including biological, social, communication, economical, and financial networks. Of particular interest is the investigation and identification of communities within such networks. In fact, identifying communities within a network provides information on how the nodes are connected and organized [1]. Especially challenging is the task of community detection within financial time series data, given their temporal dependencies, inherent noise, and non-stationarity [2]. Identifying stock communities is particularly important for portfolio strategies and risk management tasks [3]. Conventional methodologies for community detection involve

transforming time series into a graph-based representation by filtering the correlation matrix [4, 5]. Common techniques encompass the utilization of a threshold, referred to as the “Asset Graph” approach [6, 7], or employing alternative embedding methods to reduce dimensionality. Notably, Random Matrix Theory (RMT) [8, 9] is a prevalent approach, which filters the correlation matrix by identifying and eliminating eigenvalues associated with noise. However, reliance on the correlation matrix for time series representation introduces biases, which can be summarized, among other constraints, by assuming temporal stationarity, and focusing on linear relationships [10].

In this study, we propose a paradigm shift by substituting the correlation matrix with a similarity matrix derived from time series signatures [11, 12]. The rationale behind employing the signature, as opposed to the original time series, lies in its remarkable capability to encapsulate temporal information of the underlying time series within a systematically structured sequence of tensors [13, 14].

The rest of the paper is organized as follows: Sect. 2 provides an overview of core filtering techniques for correlation matrices and delves into the intricacies of community detection in the context of financial time series. Section 3 summarizes the “Asset Graph” approach, RMT, community detection, and signatures. Section 4 defines the similarity matrix derived from the signature, Sect. 5 provides a numerical experimental comparison, and Sect. 6 concludes the paper.

2 Related Work

Analyzing the structural properties and filtering techniques in correlation matrices of time series data can be approached through various methods. One straightforward approach is based on the “Asset Graph”, wherein the correlation matrix is filtered using a threshold-based method [6, 7]. Specifically, matrix elements are retained if they are greater than or equal to a given threshold value. However, the challenge lies in determining the optimal threshold value. Potential solutions include considering statistically significant correlation coefficients [15] or setting the threshold by imposing that nodes within the same community have larger probabilities to be connected by edges than background edge probabilities [16].

Another filtering technique involves utilizing a Minimum Spanning Tree (MST) [17]. This method selects a subset of edges forming a tree that connects all nodes through the links associated with the largest correlation. MST has been applied to filter the correlation matrix of the top 100 capitalized stocks on the U.S. markets [18]. Building upon the MST concept, the Planar Maximally Filtered Graph (PMFG) was introduced [19], which differs by retaining $3(n - 2)$ links compared to MST’s $n - 1$ links, where n represents the number of nodes. PMFG also allows for cliques and loops. PMFG was utilized to study the New York Stock Exchange’s (NYSE) correlation matrix’s topological properties [20].

The Random Matrix Theory (RMT) [8, 9] offers another approach for filtering correlation matrices to extract meaningful information while removing noise. RMT has been employed to study eigenvalues’ density and structural properties of empirical correlation matrices, such as those of Standard and Poor’s 500

(S&P500) [21] and the Tokyo Stock Exchange [22]. Additionally, RMT has been applied to filter the Financial Times Stock Exchange (FTSE) index's correlation matrix for portfolio creation [23], to analyze eigenvalue properties and cluster stocks in the FTSE index and S&P500 [24], and for community detection in the S&P500 [25].

Other clustering techniques include the Potts method [26], which detects modules based on a dense weighted network representation of stock price correlations, and node-based clustering [27], applicable to foreign exchange data and capable of tracking temporal dynamics. The Generalized Autoregressive Conditional Heteroskedastic (GARCH) model has been used to denoise Japanese stock return correlation matrices, followed by spectral clustering [28]. Clustering based on similarity in distribution of exchange rates in the international Foreign Exchange (Forex) market has been explored [29].

Finally, the community detection problem is typically addressed using two established algorithms: the Louvain Community Detection Algorithm [30] and the Clauset-Newman-Moore Greedy Modularity Maximization Algorithm [31].

3 Preliminaries

In this section, we provide an overview of the primary methodologies employed for filtering correlation matrices. Furthermore, we define the community detection problem and present the key algorithm utilized to address it. We subsequently introduce the concept of a time series signature and expand upon its derivation from time series data.

Consider a collection of N univariate time series denoted as S , each consisting of realizations over T discrete time steps, represented as $S_i = \{s_i(1), s_i(2), \dots, s_i(T)\}$. The entries of the correlation matrix C among the N time series can be defined as follows:

$$C_{ij} = \frac{\sigma_{S_i, S_j}}{\sqrt{\sigma_{S_i}^2} \sqrt{\sigma_{S_j}^2}}. \quad (1)$$

Here, σ_{S_i, S_j} represents the covariance between time series i and j , while $\sigma_{S_i}^2$ denotes the variance of time series i . These are expressed empirically as:

$$\begin{aligned} \sigma_{S_i, S_j} &= \frac{1}{T} \sum_{t=1}^T s_i(t) s_j(t) - \frac{1}{T} \sum_{t=1}^T s_i(t) \frac{1}{T} \sum_{t=1}^T s_j(t), \\ \sigma_{S_i}^2 &= \frac{1}{T} \sum_{t=1}^T s_i^2(t) - \left(\frac{1}{T} \sum_{t=1}^T s_i(t) \right)^2. \end{aligned}$$

The correlation matrix in this study will be constructed based on the logarithmic increments of the time series S_i . Logarithmic increments are defined as $r_i(t) = \log\left(\frac{S_i(t)}{S_i(t-1)}\right)$ for all t belonging to the set $\{2, 3, \dots, T\}$.

3.1 Asset Graph

The Asset Graph is based on the utilization of a threshold-based approach for filtering the correlation matrix. Specifically, we retain those entries within the correlation matrix that are greater than or equal to a predefined threshold. The selection of an appropriate threshold can be approached through various methods. An initial method involves evaluating multiple threshold values and selecting the one that generates a sparsely filtered matrix without introducing excessive disconnected components. An alternative strategy, outlined in [15], centers on retaining only those correlation entries that possess statistical significance.

3.2 Random Matrix Theory

The objective of this approach is to extract information from the correlation matrix by discerning and retaining the relevant components while excluding the noisy elements based on the eigenvalues' distribution. Consider a correlation matrix derived from a set of N completely random time series, each with a length of T . Following the principles of RMT, when $N \rightarrow +\infty$, $T \rightarrow +\infty$, and $1 < \lim T/N < +\infty$, the eigenvalues of the correlation matrix follow the Marcenko-Pastur distribution [21, 32], denoted with $\rho(\lambda)$:

$$\rho(\lambda) = \frac{Q}{2\pi\sigma^2} \frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{\lambda} \quad \text{if } \lambda \in [\lambda_-, \lambda_+] \quad (2)$$

and zero otherwise. Here, $Q = \lim \frac{T}{N}$, $\lambda_{\pm} = \sigma^2 \left(1 \pm \sqrt{\frac{1}{Q}}\right)^2$, and σ^2 is the variance of the elements, often set empirically as $\sigma^2 = 1 - \frac{\lambda_{max}}{N}$, where λ_{max} represents the maximum eigenvalue of the correlation matrix. In RMT, eigenvalues greater than λ_+ are statistically significant, while the rest are largely attributable to random noise. As such, any correlation matrix can be decomposed into the sum of a structural component $C^{(s)}$ comprising eigenvalues above λ_+ , and a noise component $C^{(r)}$ which can be expressed as:

$$C^{(r)} = \sum_{i:\lambda_i \leq \lambda_+} \lambda_i v_i v_i^\dagger. \quad (3)$$

Here, v_i represents the eigenvector associated with eigenvalue λ_i , and v_i^\dagger is its conjugate transpose.

However, in case of the empirical log-returns correlation matrix for N stocks, an eigenvalue often is significantly greater than the rest, and is commonly referred to as the ‘‘market mode’’ [5, 8, 21]. The market mode encapsulates the market’s overall behavior, impacting all other stocks. Consequently, removing the market mode is essential for enhancing the detection of valuable correlations by reducing noise interference. The correlation matrix for N stocks can thus be decomposed into three components:

$$C = C^{(r)} + C^{(m)} + C^{(g)}, \quad (4)$$

where $C^{(r)}$ represents the noise component, $C^{(m)}$ is the market component, and $C^{(g)}$ is the remaining significant correlation, after the removal of noise and market mode components. Specifically:

$$C^{(m)} = \lambda_{max} v_{max} v_{max}^\dagger, \quad (5)$$

$$C^{(g)} = \sum_{i: \lambda_+ < \lambda_i < \lambda_{max}} \lambda_i v_i v_i^\dagger, \quad (6)$$

where v_{max} represents the eigenvector associated with the maximum eigenvalue of the correlation matrix. Finally, the filtered correlation matrix utilized for the community detection problem (see the next subsection) is $C^{(g)}$.

3.3 Community Detection

Community detection aims to identify groups of nodes within a network that are more likely to be interconnected among themselves than with nodes from other communities [1, 33]. For the identification of non-overlapping communities, we employ the modularity optimization approach [35], chosen for its foundation in defining a null model that serves as a reference to evaluate the network's structure. Modularity acts as a metric to assess the quality of the identified partition. Indeed, partitions with high modularity have, respectively, dense/sparse connections of nodes within/between their clusters.

Consider a network with N nodes and the associated adjacency matrix $A \in \mathbb{R}^{N \times N}$. In the context of an undirected unweighted graph, the entries of the adjacency matrix, a_{ij} , are such that $a_{ij} = 1$ if a link exists between nodes i and j , and 0 otherwise. Our goal is to find non-overlapping communities represented by an N -dimensional vector η , where the i -th component η_i indicates the set to which node i belongs, as defined in [25]. The modularity, denoted as $Q(\eta)$, is defined as follows:

$$Q(\eta) = \frac{1}{A_{tot}} \sum_{i,j} [A_{ij} - \langle A_{ij} \rangle] \delta(\eta_i, \eta_j), \quad (7)$$

where $\delta(\eta_i, \eta_j)$ is the Kronecker delta function equal to 1 if $\eta_i = \eta_j$, and 0 otherwise, signifying that only nodes within the same community are considered. $A_{tot} = \sum_{i,j} A_{ij} = 2l$ is twice the total number of links l , and $\langle A_{ij} \rangle$ represents the employed null model. Traditionally, it is the so-called configuration model, in which $\langle A_{ij} \rangle = \frac{k_i k_j}{2l}$, with k_i as the degree of node i [33].

In the presence of finite time series data having a global mode in the correlation matrix, the modularity can be expressed as follows:

$$Q(\eta) = \frac{1}{C_{norm}} \sum_{i,j} [C_{ij} - C_{ij}^{(r)} - C_{ij}^{(m)}] \delta(\eta_i, \eta_j) = \frac{1}{C_{norm}} \sum_{i,j} C_{ij}^{(g)} \delta(\eta_i, \eta_j) \quad (8)$$

where $C^{(r)}$, $C^{(m)}$, and $C^{(g)}$ represent the noise, market, and significant correlation components, respectively, as defined in equations (3), (5), and (6). Additionally, $C_{norm} = \sum_{i,j} C_{ij}$. Research such as [22, 23, 34] has demonstrated that

the eigenvector components of $C^{(g)}$ exhibit alternating signs, allowing for the identification of groups of stocks influenced by similar factors. This provides an effective basis for community detection in financial networks.

The modularity $Q(\eta)$ assumes values within the interval $[-0.5, 1]$, indicating the edge density within communities relative to edges between communities. Higher modularity values denote a stronger community structure, with nodes forming distinct clusters, while lower values suggest a more uniform distribution of edges across the network.

3.4 Signature

The concept of signature derives from the field of path theory, providing a structured and comprehensive representation of the temporal evolution within a time series. Its potency lies in capturing both temporal and geometric patterns embedded within the time series. Temporal patterns encompass long-term dependencies and recurrent trends across time, while geometric patterns encompass the shape of time series trajectories, and intricate data behaviors such as loops and self-intersections [14].

For the sake of clarity, we shall adopt the notation presented in [36] and restrict our discussion to continuous functions mapping from a compact time interval $J := [a, b]$ to \mathbb{R}^d with finite p -variation, all starting from the origin. This space is denoted as $C_0^p(J, \mathbb{R}^d)$.

Let $T((\mathbb{R}^d)) := \bigoplus_{k=0}^{\infty} (\mathbb{R}^d)^{\otimes k}$ represent a tensor algebra space, encompassing the signatures of \mathbb{R}^d -valued paths, offering their comprehensive representation. Furthermore, let $S_i = \{s_i(1), s_i(2), \dots, s_i(T)\}$ denote a discrete time series. To bridge the gap between the discrete and continuous cases, the time series needs to be converted into a continuous path, achieved through methods like the lead-lag transformation or the time-join transformation [37]. Let L be the continuous path produced by the lead-lag transformation, which we adopt due to its ability to directly extract various features including path volatility (which comes from the second term of the signature), a crucial facet in finance. Consequently, we define the signature \mathcal{S} and the truncated signature at level M , denoted as \mathcal{S}_M , as follows:

Definition 1 (Signature and Truncated Signature). *Let $L \in C_0^p(J, \mathbb{R}^d)$ be a path. The signature \mathcal{S} of the path L is defined as:*

$$\mathcal{S} = (1, L_J^1, \dots, L_J^k, \dots) \in T((\mathbb{R}^d)), \quad (9)$$

where $L_J^k = \int_{t_1 < t_2 < \dots < t_k, t_1, \dots, t_k \in J} dL_{t_1} \otimes \dots \otimes dL_{t_k}$ are called iterated integrals.

The truncated signature of degree M is defined as:

$$\mathcal{S}_M = (1, L_J^1, \dots, L_J^M). \quad (10)$$

The signature structure offers a hierarchical interpretation, with lower-order components capturing broad path attributes and higher-order terms revealing intricate characteristics (including higher-order moments, and local geometric

features). Importantly, the signature remains invariant under reparameterization, preserving integral values despite time transformations. It also adheres to translation invariance and concatenation properties [38]. The truncated signature preserves the first $\frac{d^{M+1}-1}{d-1}$ iterated integrals, with M denoting truncation degree and d representing path dimension. The factorial decay of neglected iterated integrals ensures minimal information loss in truncation of \mathcal{S} [39].

Given two stochastic processes, A and B , defined on a probability space $(\Omega, \mathbb{P}, \mathcal{F})$, and supposing equation (9) holds almost surely for both A and B , with expected values of $\mathcal{S}(A)$ and $\mathcal{S}(B)$ being finite, we have the following theorem [40]:

Theorem 1 (Expected Signature). *Let A and B be two $C_0^1(J, \mathbb{R}^d)$ -valued random variables. If $\mathbb{E}[\mathcal{S}(A)] = \mathbb{E}[\mathcal{S}(B)]$, and $\mathbb{E}[\mathcal{S}(A)]$ has infinite radius of convergence, then $A \stackrel{d}{=} B$, i.e., A and B are equal in distribution.*

The signature uniquely defines a path's trajectory [11], under suitable assumptions, while the expected signatures uniquely determine the distributions of paths, paralleling the role of moment generating functions [41]. For a more comprehensive elaboration, rigorous formulations, and visual examples, consult [14, 37, 42].

4 Signature-Based Similarity Matrix

In our research, we introduce a novel approach that replaces the conventional correlation matrix, C , with a similarity matrix derived from the truncated signature of the logarithmic increments of each time series S_i . This novel concept is rooted in the uniqueness of the signature, which can be likened to the moment generating function, making it an ideal candidate for quantifying similarity between time series. The hypothesis here is that if two time series possess highly similar signatures, they should exhibit substantial similarity in their behaviors.

To construct this similarity matrix, we embark on a multi-step process. First, we apply the lead-lag transformation to the logarithmic increments of each time series S_i , yielding the path L_i . Subsequently, we compute the truncated signature, denoted as \mathcal{S}_M , with a truncation degree M set to 3, applied to L_i , also denoted by $\mathcal{S}_M(L_i)$. We then proceed to compute a similarity measure between each pair of stocks based on their truncated signatures. Three distinct measures are employed for this purpose: Euclidean Distance (ED), Cosine Similarity (CS), and Radial Basis Function (RBF) kernel. The choice of these measures is deliberate: ED is selected for its sensitivity to data magnitude and computational efficiency, CS for its scale invariance, resistance to outliers, and suitability for time series trend analysis, and the RBF kernel for its capability to capture complex non-linear relationships. Moreover, we convert the Euclidean distance (which is a dissimilarity metric) into a similarity metric using a strictly monotone decreasing function, specifically $f(x) = \frac{1}{a+x}$ with $a = 1$.

Finally, we obtain a similarity matrix P , with entries $p_{ij} \in [0, 1]$, where a value of 1 signifies that elements i and j are perfectly alike. Following the creation of this similarity matrix, we subject it to filtering processes, specifically using

the threshold method and RMT, as elaborated upon in Sect. 3. This filtering serves the purpose of retaining only the significant similarities among the time series, thereby enhancing the robustness and effectiveness of our approach.

Table 1. Modularity results for the correlation matrix and for the signature-based similarity matrix for the Standard and Poor’s 500.

Data Type	Filtering Method	Modularity (Louvain)	Modularity (Greedy)	# Cluster (Louvain)	# Cluster (Greedy)
Correlation Matrix	Threshold	0.0207	0.0159	81	11
	RMT	0.0987	0.1185	4	11
Signature-based Similarity Matrix	Threshold_ED	0.1796	0.1783	8	9
	Threshold_CS	0.0020	-0.0134	430	11
	Threshold_RBF	0.0976	0.0952	112	11
	RMT_ED	0.1975	0.1994	5	11
	RMT_CS	0.1177	0.8527	2	11
	RMT_RBF	0.1326	0.1428	4	11

5 Experimental Evaluation

The principal aim of this study is to showcase a substantial enhancement in the modularity metric when replacing the traditional correlation matrix with a similarity matrix derived from signatures in the context of community detection. A higher modularity score signifies an improved capability of the algorithm to identify more cohesive and distinguishable communities in the dataset. Additionally, our investigation highlights that the identified communities do not rigidly align with the initially assigned data categories.

To conduct this analysis, we focus our attention to the S&P500 stock exchange, a market encompassing 500 major publicly traded companies spanning diverse sectors and industries in the United States market [43]. Notably, this index classifies each stock into one of eleven distinct sectors: Communication Services, Consumer Discretionary, Consumer Staples, Energy, Financials, Health Care, Industrials, Information Technology, Materials, Real Estate, and Utilities. Our data collection process covers the period from Saturday 10th July, 2010 to Monday 10th July, 2023. After computing the logarithmic returns and eliminating stocks with insufficient data, our dataset encompasses 443 stocks for analysis, each comprising 3720 observations. Consequently, we denote $N = 443$ as the number of stocks and $T = 3720$ as the length of each time series.

In this study, we use as baseline models the correlation matrix filter with a predefined threshold and the RMT-based filter, as detailed in Sect. 3. Within the threshold method, we determine the optimal threshold value following the procedure in [25]. Subsequently, we retain correlation entries exceeding the threshold of 0.0437. The eigenvalue distribution under the RMT framework confirms the

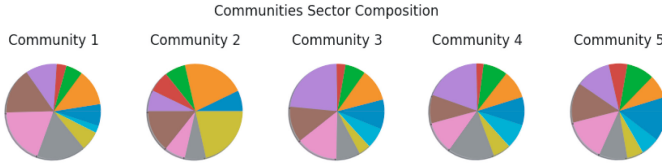


Fig. 1. Communities identified using the Louvain Community Detection Algorithm on the signature-based Similarity Matrix obtained via the Euclidean Distance and subsequently filtered using RMT. Communication Services (■), Consumer Discretionary (■), Consumer Staples (■), Energy (■), Financials (■), Health Care (■), Industrials (■), Information Technology (■), Materials (■), Real Estate (■), and Utilities (■).

presence of the market model. Specifically, the largest eigenvalue of the correlation matrix is approximately 174, with the second-largest eigenvalue around 20. Consequently, we apply the filtering process to the correlation matrix as defined in equation (6). Notably, the market mode is also observed in the similarity matrix based on signatures.

Table 1 presents the modularity results for the analyzed models, utilizing both the Louvain Community Detection Algorithm and the Clauset-Newman-Moore Greedy Modularity Maximization Algorithm. Utilizing a signature-derived similarity matrix consistently yields higher modularity values, indicating superior performance in identifying more distinct communities than the correlation matrix approach. The exception is when applying threshold-based filtering to the cosine similarity matrix. Furthermore, results are more consistent when using similarity matrices based on the Euclidean distance, whereas outcomes from the cosine similarity matrix vary significantly in terms of modularity values and cluster count.

Remarkably, the Louvain algorithm, when applied to the threshold-filtered matrix, identifies numerous smaller communities, each comprising a single stock. This phenomenon holds for both the correlation matrix and the signature-based similarity matrix, except for the Euclidean distance case.

Figure 1 illustrates the community structure generated by the Euclidean distance-based similarity matrix, filtered using RMT. The algorithm identifies a total of 5 communities, in contrast to the S&P500 index that classifies these stocks into 11 distinct categories. All 5 algorithm-identified communities encompass stocks from different sectors according to the S&P500 classification, suggesting concealed correlations among stocks from various sectors, highlighting intricate inter-sector relationships.

To evaluate the robustness of our proposed methodology, we conducted a stability analysis. This entailed gradually increasing the number of observations in the dataset, starting with roughly one-third of the original observations for the 443 stocks. We incrementally added observations until reaching the dimensions of the original dataset. This analysis aimed to demonstrate that our method's

effectiveness remains consistent regardless of the quantity of observations considered. It is important to note that RMT requires only that the number of observations (T) exceeds the number of stocks (N). The results of this stability analysis are presented in Fig. 2. In this study, we primarily examine the similarity matrix derived from the Euclidean distance via the signature, given its previously demonstrated superior performance in achieving higher modularity. We also include the conventional correlation matrix in our analysis, subjecting both matrices to RMT-based filtration. Interestingly, even within this context, the modularity consistently exhibits higher values when utilizing the signature-based similarity matrix for community detection. Notably, the performance of the greedy algorithm for community detection within the signature-based approach appears to be influenced by the volume of observations.

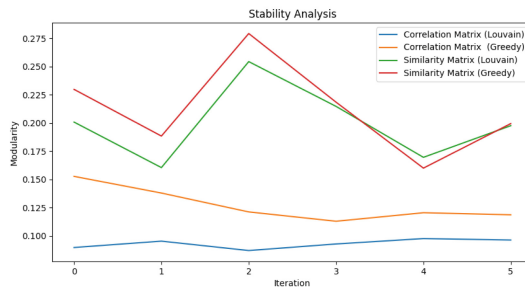


Fig. 2. Stability analysis for the correlation matrix and the signature-based similarity matrix for the Standard and Poor’s 500.

6 Conclusion

This study explores contemporary techniques for filtering correlation matrices in community detection. We introduce a novel approach, substituting the correlation matrix with a signature-derived similarity matrix. We evaluate three similarity measures: nonlinearly transformed Euclidean distance, cosine similarity, and Radial Basis Function (RBF) similarity. Using the S&P500 dataset, we empirically assess this method’s performance with the modularity metric. Results consistently indicate enhanced modularity, with the Euclidean distance-based similarity matrix performing the best.

Future research will involve in-depth exploration of the structural properties in community detection using the signature-based similarity matrix. In particular, we plan to: investigate the reasons behind the higher modularity achieved by the proposed method with respect to other similarity measures; extend it by directly computing the signature of the vector of (paths derived from) time series; explore its integration into portfolio optimization and risk management strategies; apply it to other contexts involving time series, such as movement analysis.

Acknowledgment. The authors were partially supported by the PRIN 2022 project “Multiscale Analysis of Human and Artificial Trajectories: Models and Applications”, funded by the European Union - Next Generation EU program (CUP: D53D23008790006).

References

1. Albert-László, B.: Network science. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371.1987, article no. 20120375 (2013)
2. Tsay, R.S.: *Analysis of Financial Time Series*. 2nd edn. John Wiley & Sons (2005)
3. Prigent, J. L.: *Portfolio optimization and performance analysis*. CRC Press (2007)
4. Mantegna, R.N., Stanley, H. E.: *Introduction to econophysics: correlations and complexity in finance*. Cambridge University Press (1999)
5. Sinha, S., Chatterjee, A., Chakraborti, A., Chakrabarti, B. K.: *Econophysics: An Introduction*. John Wiley & Sons (2010)
6. Onnela, J.P., Kaski, K., Kertész, J.: Clustering and information in correlation based financial networks. In: *The European Physical Journal B*, vol. 38, pp. 353–362 (2004)
7. Heimo, T., Kaski, K., Saramäki, J.: Maximal spanning trees, asset graphs and random matrix denoising in the analysis of dynamics of financial networks. In: *Physica A: Statistical Mechanics and its Applications*, vol. 388(2–3), pp. 145–156 (2009)
8. Mehta, M. L.: *Random matrices*. Elsevier (2004)
9. Bai, Z., Silverstein, J. W.: *Spectral analysis of large dimensional random matrices* (Vol. 20). New York: Springer (2010). <https://doi.org/10.1007/978-1-4419-0661-8>
10. Brockwell, P.J., Davis, R.A.: *Introduction to Time Series and Forecasting*, 2nd edn. Springer, New York, New York, NY (2002). <https://doi.org/10.1007/b97391>
11. Lyons, T. J.: Differential equations driven by rough signals. In: *Revista Matemática Iberoamericana*, vol. 14(2), pp. 215–310. (1998)
12. Lyons, T., Ni, H., Oberhauser, H.: A feature set for streams and an application to high-frequency financial tick data. In: *In Proceedings of the 2014 International Conference on Big Data Science and Computing*, pp. 1–8 (2014)
13. Chen, K.T.: Integration of paths A faithful representation of paths by noncommutative formal power series. In: *Transactions of the American Mathematical Society*, vol. 89(2), pp. 395–407 (1958)
14. Lyons, T.: Rough paths, signatures and the modelling of functions on streams (2014). In: *arXiv preprint arXiv:1405.4537*
15. Fisher, R.A.: Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. In: *Biometrika*, vol. 10(4), pp. 507–521 (1915)
16. Yuan, S., Wang, C., Jiang, Q., Ma, J.: Community detection with graph neural network using Markov stability. In: *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIC)*, pp. 437–442. IEEE (2022)
17. Mantegna, R.N.: Hierarchical structure in financial markets. In: *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 11, pp. 193–197 (1999)

18. Bonanno, G., Caldarelli, G., Lillo, F., Micciche, S., Vandewalle, N., Mantegna, R. N.: Networks of equities in financial markets. In: *The European Physical Journal B*, vol. 38, pp. 363–371 (2004)
19. Tumminello, M., Aste, T., Di Matteo, T., Mantegna, R.N.: A tool for filtering information in complex systems. In: *Proceedings of the National Academy of Sciences*, vol. 102(30), pp. 10421–10426 (2005)
20. Tumminello, M., Di Matteo, T., Aste, T., Mantegna, R. N.: Correlation based networks of equity returns sampled at different time horizons. In: *The European Physical Journal B*, vol. 55, pp. 209–217 (2007)
21. Laloux, L., Cizeau, P., Bouchaud, J.P., Potters, M.: Noise dressing of financial correlation matrices. In: *Physical Review Letters*, vol. 83(7), pp. 1467 (1999)
22. Utsugi, A., Ino, K., Oshikawa, M.: Random matrix theory analysis of cross correlations in financial markets. In: *Physical Review E*, vol. 70(2), pp. 026110 (2004)
23. Potters, M., Bouchaud, J.P., Laloux, L.: Financial applications of random matrix theory: Old laces and new pieces (2005). arXiv preprint physics/0507111
24. Livan, G., Alfarano, S., Scalas, E.: Fine structure of spectral properties for random correlation matrices: an application to financial markets. In: *Physical Review E*, vol. 84(1), pp. 016113 (2011)
25. MacMahon, M., Garlaschelli, D.: Community detection for correlation matrices. In: *Physical Review X*, vol. 5(11), pp. 021006 (2015)
26. Heimo, T., Kumpula, J.M., Kaski, K., Saramäki, J.: Detecting modules in dense weighted networks with the Potts method. In: *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008(08), pp. P08007 (2008)
27. Fenn, D.J., et al.: Dynamical clustering of exchange rates. In: *Quantitative Finance*, vol. 12(10), pp. 1493–1520 (2012)
28. Isogai, T.: Clustering of Japanese stock returns by recursive modularity optimization for efficient portfolio diversification. In: *Journal of Complex Networks*, vol. 2(4), pp. 557–584 (2014)
29. Chakraborty, A., Easwaran, S., Sinha, S.: Uncovering the hierarchical structure of the international Forex market by using similarity metric between the fluctuation distributions of currencies (2020). In: arXiv preprint [arXiv:2005.02482](https://arxiv.org/abs/2005.02482)
30. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. In: *Journal of statistical mechanics: theory and experiment*, vol. 2008(10), pp. P10008 (2008)
31. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. In: *Physical review E*, vol. 70(6), pp. 066111 (2004)
32. Plerou, V., Gopikrishnan, P., Rosenow, B., Amaral, L.A.N., Stanley, H.E.: Universal and nonuniversal properties of cross correlations in financial time series. In: *Physical Review Letters*, vol. 83(7), pp. 1471 (1999)
33. Fortunato, S.: Community detection in graphs. In: *Physics reports*, vol. 486(3–5), pp. 75–174 (2010)
34. Plerou, V., Gopikrishnan, P., Rosenow, B., Amaral, L.A.N., Guhr, T., Stanley, H.E.: Random matrix approach to cross correlations in financial data. In: *Physical Review E*, vol. 65(6), pp. 066126 (2002)
35. Newman, M. E., Girvan, M.: Finding and evaluating community structure in networks. In: *Physical Review E*, vol. 9(2), pp. 026113 (2004)
36. Ni, H., Szpruch, L., Wiese, M., Liao, S., Xiao, B.: Conditional sig-wasserstein GANs for time series generation (2020). In: arXiv preprint [arXiv:2006.05421](https://arxiv.org/abs/2006.05421)
37. Levin, D., Lyons, T., Ni, H.: Learning from the past, predicting the statistics for the future, learning an evolving system (2013). In: arXiv preprint [arXiv:1309.0260](https://arxiv.org/abs/1309.0260)

38. Chen, K.T.: Integration of paths—A faithful representation of paths by noncommutative formal power series. In: Transactions of the American Mathematical Society, vol. 89(2), pp. 395–407 (1958)
39. Lemercier, M., Salvi, C., Damoulas, T., Bonilla, E., Lyons, T.: Distribution regression for sequential data. In: In International Conference on Artificial Intelligence and Statistics, pp. 3754–3762. PMLR (2021)
40. Lyons, T., Ni, H.: Expected signature of Brownian motion up to the first exit time from a bounded domain. In: The Annals of Probability, vol. 43(5), pp. 2729–2762 (2015)
41. Chevyrev, I., Lyons, T.: Characteristic functions of measures on geometric rough paths. In: The Annals of Probability, vol. 44(6), pp. 4049–4082 (2016)
42. Chevyrev, I., Kormilitzin, A.: A primer on the signature method in machine learning (2016). In: arXiv preprint [arXiv:1603.03788](https://arxiv.org/abs/1603.03788)
43. S&P Global Homepage. <https://www.spglobal.com>



Hierarchical Overlapping Community Detection for Weighted Networks

Petr Prokop^(✉), Pavla Dráždilová, and Jan Platoš

Department of Computer Science, VSB - Technical University of Ostrava,
708 00 Ostrava-Poruba, Czech Republic
{petr.prokop,pavla.drazdilova,jan.platos}@vsb.cz

Abstract. Real-world networks often contain community structures, where nodes form tightly interconnected clusters. Recent research indicates hierarchical organization, where vertices split into groups that further subdivide across multiple scales. However, individuals in social networks typically belong to multiple communities due to their various affiliations, such as family, friends, and colleagues. These overlaps will emerge in the community structure of online social networks and other complex networks like in biology, where nodes have diverse functions. In this work, we propose an algorithm for hierarchical overlapping community detection in weighted networks. The overlap between clusters is realized via maximal cliques that are used as base elements for hierarchical agglomerative clustering on the graph (GHAC). The closed trail distance and the size of the maximal clique in overlap are used for the dissimilarity between clusters in agglomerative steps of the GHAC. The closed trail distance is designed for weighted networks.

Experiments on synthetic networks and different evaluations of the results of experiments show that the proposed algorithm is comparable with other widely used algorithms for overlapping community detection and is efficient for detecting hierarchy structure in weighted networks.

Keywords: overlapping communities · closed trail distance · weighted graph · hierarchical structure

1 Introduction

Most real networks are characterized by overlapped and nested communities [25]. Modular and hierarchical community structures are ubiquitous in complex real-world networks. The observation that large clusters consist of smaller clusters of higher quality can be evidence of hierarchical structure in the network, which is characteristic of many complex systems [37]. Hierarchical structure goes beyond simple clustering and explicitly includes organization at all scales in a network simultaneously. Conventionally, the hierarchical structure can be represented by a tree (dendrogram). Clauset et al. [9] show that knowledge of hierarchical structure can be used to predict missing connections in partially known networks with high accuracy. Their results suggest that hierarchy is a central organizing principle of complex networks, capable of offering insight into many network

phenomena. The algorithms for the detection of the hierarchical structure of communities were proposed in [9, 17, 22, 34, 35, 47].

The hierarchical agglomerative methods for community detection use some distance or similarity between nodes for dissimilarity or similarity calculation between clusters. The authors in the article [8] use modified structural similarity (cosine similarity) in HAC, the extended Jaccard similarity is used in the article [4], and the Wasserman-Faust distance [42] was successfully used by Brzozowski et al. in [7]. Dissimilarity used in the proposed algorithm is based on the closed trail distance (*CT*-distance) [38] between nodes u, v that corresponds in the undirected graph to the length of the shortest closed trail containing nodes u, v . The distance between nodes in the network, designed in this way, takes into account the structure of the graph and reflects the density of the neighborhood of the nodes [39]. The closed trail distance designed for weighted networks (*wCT*-distance) prefers such a closed trail that has the smallest weight. It can be achieved by a shorter length or a smaller cost of edges in the closed trail.

The presence of overlapping communities in complex networks reflects the multifaceted nature of interactions among nodes. Overlapping communities indicate that nodes can belong to more than one group simultaneously, capturing the patterns that often characterize real-world networks. For overlapping community detection can be used different approaches: local method [18, 32], modularity optimization [33, 36], clique based [25, 45], statistical inference based methods [18, 28], label propagation [27, 43], spectral clustering [46], and other methods.

An important part of the hierarchical community detection method is choosing the appropriate level in the hierarchical structure. We can determine the number of clusters if we focus on the structure of network communities at different mesoscopic scales [26]. On the other hand, we can choose a cut-off level such that it corresponds to the best value of the chosen evaluation of the detected overlapped communities. Modified modularities [19, 35] are used for the evaluation of overlapped communities without knowledge of ground truth.

The most popular information recovery metrics for overlapping communities with the knowledge of ground truth are the overlapping Normalized Mutual Information (ONMI) [17], the average F1-score [44], the Omega index [10]. Evaluations of detected overlapped communities are discussed in articles [11, 12, 41].

The main contributions of this paper are the introduction of a new hierarchical overlapping method for weighted networks and the empirical evaluation of community detection quality for the proposed method on synthetic networks generated by LFR. The proposed algorithm is suitable for detecting communities with higher overlaps.

This paper is organized as follows. Section 2 presents related work about overlapping and hierarchical community detection. In Sect. 3, we propose an algorithm for overlapping hierarchical community detection in the weighted network. Finally, Sect. 4 contains experiments on synthetic networks (LFR) and their results.

2 Related Work

Algorithms for hierarchical community detection can be categorized into divisive, agglomerative, or optimization-based methods. Divisive algorithms start with the entire network as a single community and recursively partition it into smaller sub-communities. Recursive spectral partitioning is applied in an article [22] for hierarchical community detection. Schaub et al. [34] introduced a definition of hierarchy based on the concept of stochastic externally equitable partitions and their relation to probabilistic models, such as the stochastic block model. They enumerate the challenges involved in detecting hierarchies and, by studying the spectral properties of hierarchical structure, present an efficient and principled method for detecting them.

Agglomerative methods usually start with each node or edge as its own community and then iteratively merge communities based on certain criteria. As the process continues, communities are combined in a hierarchical manner. Optimization-based methods formulate the hierarchical community detection problem as an optimization task and aim to find the best division of the network into communities according to specific criteria. These categories of methods for hierarchical community detection are not always distinct, and some algorithms might incorporate elements from multiple approaches. The Louvain method [5] performs a greedy optimization in an agglomerative hierarchical manner, by assigning each vertex to the community of their neighbors yielding the largest modularity, and creating a smaller weighted super-network whose vertices are the clusters found previously. Partitions found on this super-network hence consist of clusters including the ones found earlier, and represent a higher hierarchical level of clustering. The procedure is repeated until one reaches the level with the largest modularity.

A lot of methods for overlapping community detection in complex networks are based on cliques that represent the densest subgraphs. The local clique merging algorithm (LCMA) presented by Li et al. [23] is one of the represents of this approach. The other representative is the clique percolation method (CPM) by Palla et al. [25] used in the tool CFinder [1]. CPM defines a community as a union of all k -cliques that can be reached by each other through a series of adjacent k -cliques (two k -cliques are adjacent if they share exactly $k - 1$ vertices). Extension of CPM on weighted networks has been presented by Farkas et al. in [13]. The Greedy Clique Expansion (GCE) algorithm proposed by Lee et al. [20] identifies distinct cliques as seeds and expands these seeds by greedily optimizing a local fitness function. In [15], Gupta et al. reviewed a lot of clique-based overlapping community detection algorithms.

Some methods for overlapping community detection are based on density peak clustering (DPCLUS). DPCLUS method was introduced in the article [3] where the algorithm identifies subgraphs that satisfy a density condition and certain cluster connectivity properties. One of the first modifications of DPCLUS is in [21] where the method IPCA looks for subgraph structures that have a small diameter (or a small average vertex distance) and satisfy a different cluster connectivity-density property. DPCLUS was used in an article [24] to obtain the

number of centers as the pre-defined parameter for nonnegative matrix factorization.

Lancichinetti et al. proposed in [17] the algorithm that finds both overlapping communities and the hierarchical structure. Their method is based on the local optimization of a fitness function that is calculated from the internal and external degrees of the nodes of the module. In the same year was published the article [35] where Shen et al. presented the algorithm (EAGLE) to detect both the overlapping and hierarchical properties of complex community structures together. EAGLE deals with the set of maximal cliques and adopts an agglomerative framework. The similarity between clusters is based on modularity. The authors propose their extension of modularity to evaluate the goodness of overlapped community decomposition. Algorithms GCE [20] and EAGLE [35] identify the maximal cliques in the network as the seeds. The results of [2] suggest that link communities are the basic building blocks that reveal overlap and hierarchical organization in networks as two aspects of the same phenomenon.

Lancichinetti et al. presented in [18] Order Statistics Local Optimization Method (OSLOM), the first method capable of detecting clusters in networks accounting for edge directions, edge weights, overlapping communities, hierarchies, and community dynamics. It is based on the local optimization of a fitness function expressing the statistical significance of clusters with respect to random fluctuations, which is estimated with tools of Extreme and Order Statistics.

The COPRA algorithm proposed by Gregory [14] is an extended Label Propagation Algorithm (LPA) [27] algorithm. It associates every label in the label set with a belonging coefficient and updates the label set of each node by averaging the coefficients from all the neighbor nodes. This algorithm detects overlapping communities and can also handle weighted and bipartite networks. The SLPA algorithm proposed by Xie et al. [43] is the other algorithm for overlapping community detection that is an extension of the LPA. It spreads the labels between nodes according to a speaker-listener pairwise interaction rule.

Zhao et al. [47] proposed a parameter-free algorithm called agglomerative clustering based on label propagation algorithm (ACLPA) to detect both overlapping and hierarchical community structures in complex networks. Authors combining the advantages of agglomerative clustering and label propagation and their algorithm can build the hierarchical tree of overlapping communities in large-scale networks.

3 Proposed Algorithm for Overlapping Hierarchical Weighted Community Detection

Hierarchical agglomerative clustering on the graph (GHAC) is used to detect nested communities in a network using novel dissimilarity among clusters based on closed trail distance in the graph and overlap density. The weight of edges is incorporated into the calculation of wCT -distance [38] that together with an overlap of cliques determines dissimilarity between clusters during the GHAC. Maximal cliques are the base elements of the GHAC. This core idea has been

presented in our previous work (under the review process) and the main contribution of this paper is the extension of the community detection method to weighted networks.

3.1 CT–distance in Weighted 2-edge-connected Graph

The definition of the CT–distance [38] (the shortest closed trail containing u and v) may be extended for the weighted graph $G = (V, E, w)$ where w is a mapping $w : E \rightarrow R^+$. The weights of edges are considered like similarity in the sense that a greater value is better. We need the reciprocal value of $w(e)$ to express the dissimilarity between vertices such that the weight of the closed trail will be determined more by weights than by length.

Definition 1. Let $G = (V, E, w)$ be a weighted graph and let the mapping $d_{wCT} : V \times V \rightarrow R_0^+$ be defined by the equation

$$d_{wCT}(u, v) = \min_{CT(u,v) \subseteq G} \left(\sum_{\forall e \in CT(u,v)} \frac{1}{w(e)} \right).$$

Then the function d_{wCT} is called the weighted closed trail distance (wCT –distance).

We introduce dissimilarities between clusters based on the wCT –distance, and the weight of overlap in a weighted graph. We consider that the weight of overlap will be represented by the densest part of overlap which is a clique. The weight of the densest clique in the overlap of two communities can be formalized by the biggest and the heaviest clique as:

$$\bar{w}(Q) = |Q| + \frac{\sum_{\forall e \in Q} w(e)}{\sum_{\forall e \in E} w(e)}.$$

We define dissimilarities based on the Complete Linkage (CL) and the Average Linkage (AL) approach for GHAC on the weighted graph $G = (V, E, w)$ as:

$$d_{wGHAC}^{CL}(C_i, C_j) = \frac{\max_{(v_i \in C_i \setminus C_j), (v_j \in C_j \setminus C_i)} d_{wCT}(v_i, v_j)}{1 + \operatorname{argmax}_{Q \in C_i \cap C_j} \bar{w}(Q)},$$

and

$$d_{wGHAC}^{AL}(C_i, C_j) = \frac{\sum_{(v_i \in C_i \setminus C_j), (v_j \in C_j \setminus C_i)} d_{wCT}(v_i, v_j)}{|(C_i \cup C_j) \setminus (C_i \cap C_j)| (1 + \operatorname{argmax}_{Q \in C_i \cap C_j} \bar{w}(Q))}.$$

For the current work, we have denoted the use of the GHAC method with dissimilarity d_{wGHAC}^{AL} as wAL GHAC (Average linkage hierarchical clustering on the weighted graph) or for dissimilarity d_{wGHAC}^{CL} as wCL GHAC.

3.2 Community Detection Procedure

A brief description of the steps in the proposed community detection method is given in Algorithm 1. Suurballe’s algorithm [40] is used to calculate wCT -distances among vertices. These distances represent one component of dissimilarity utilized in the GHAC. The size of the overlap represents the other part of dissimilarity as explained in the previous section.

- Input** : The 2-edge-connected component of a network (i.e. graph without bridges)
- Output**: Network cover
- Step 1**: Calculate wCT -distance matrix among vertices in a input graph.
- Step 2**: Find maximal cliques (Bron-Kerbosch alg. [6]).
- Step 3**: Hierarchical agglomerative clustering on the graph:
- Step 3.1**: Agglomerate communities according to proposed dissimilarity with maximal cliques as base elements.
 - Step 3.2**: Map merged clusters of base elements to origin graph vertices.
 - Step 3.3**: Filter out small clusters and fill in the network cover.
 - Step 3.4**: Evaluate network cover structural quality by modularity and external evaluation against ground truth.
 - Step 3.5**: Repeat the algorithm from Step 3.1 until all clusters are merged.
- Step 4**: Choose the best level for a cut of a dendrogram of agglomerative steps.
- Algorithm 1**: Proposed community detection method based on the GHAC and dissimilarity leveraging wCT -distance.

Maximal cliques are used as bases in the GHAC and merged clusters of maximal cliques are mapped to vertices with a few post-processing steps. The communities with a size less than 5 are removed from network cover during the post-processing. Any vertices that do not belong to any community are assigned to one of the most frequent communities among its neighbors.

The modularity defined in [19] is used to evaluate each level of a cut in the GHAC. This modularity takes the weight of edges into account. The evaluation of network cover against ground truth information is measured using ONMI [17] and NF1 [29,31] scores.

4 Experiments

The experiments are aimed at verifying the ability of the proposed and standard algorithms to correctly detect communities with large overlaps and with nodes in multiple communities. The parameters of the LFR [16] benchmark were selected depending on this requirement.

For the performance evaluation, a collection of synthetic graphs was generated with combinations of parameters for the LFR benchmark: 500 nodes, the power law exponent for degree distribution was -2 , the power law exponent for community size distribution was -1 , average and maximum degree

$(\langle k \rangle, k_{max}) \in \{(10, 30), (20, 30), (10, 50), (20, 50), (30, 50)\}$, minimum and maximum community size $(c_{min}, c_{max}) \in \{(7, 30), (15, 50)\}$, a number of nodes in overlaps $on \in \{0, 50, 100, 200\}$, a number of memberships of the overlapping nodes $om \in \{2, 4, 6\}$ and mixing parameter $\mu \in \{0.1, 0.2, 0.3\}$. Five different graph instances were generated using a unique seed for each combination of parameters. Two subsets of generated networks are used in experiments and visualized: subset A contains 540 random graphs with mixing parameter $\mu = 0.1$, and subset B selects 945 networks with “reasonable” overlap, i.e. the non-overlapping networks and networks with 40% of overlapping nodes and a number of membership $om \geq 4$ were excluded.

The selection of overlapping community detection algorithms for comparison consists of OSLOM [18], IPCA [21], SLPA [43], and ASLPAW [30]. The SLPA is the only algorithm that is not able to work with weights in networks. During the experiment, several input parameter configurations were used for each method and only the best achieved evaluation score is reported in the figures for methods. The symbol star * is used when reporting these values. In addition to a selection of possibly the best evaluation score, the score for the proposed methods is also reported only for the cut indicated by the highest overlapped modularity value. In the experiments, we are progressing from unweighted performance evaluation to performance evaluation on weighted networks.

4.1 Unweighted Synthetic Networks

The version of LFR for unweighted and undirected graphs with overlapping communities was used to assess the quality when the weight of all edges was equal to one.

The evaluation of the performance in relation to a number of nodes in overlapping communities is visualized in Fig. 1, where the average evaluation score for various benchmark configurations is reported. Non-overlapping results suggest a comparable average score for the proposed methods and SLPA and OSLOM. With the increasing number of overlapping nodes, the average score decreases for all methods. We can notice that the OSLOM achieved a better ONMI score for a higher number of overlapping nodes in comparison to the proposed method when we used modularity for selection (AL GHAC). However, the maximum ONMI score achieved in the hierarchical process (AL GHAC*) is higher. It shows a limit of used modularity in the identification of the best possible cut in the hierarchy. When the NF1 was used for evaluation the OSLOM outperformed all methods.

In the denser networks, Fig. 2, the proposed methods achieved better ONMI scores than other methods. Even the cut indicated by modularity achieved a score comparable to OSLOM. OSLOM finished with the highest values for each category of network with different densities according to the NF1 evaluation.

The decreasing trend of ONMI with increasing mixing parameter μ can be seen in Fig. 3. The proposed methods seem to be affected in a comparable way to methods SLPA and ASLPAW. The increasing mixing parameter has the smallest impact on the evaluation score of the OSLOM.

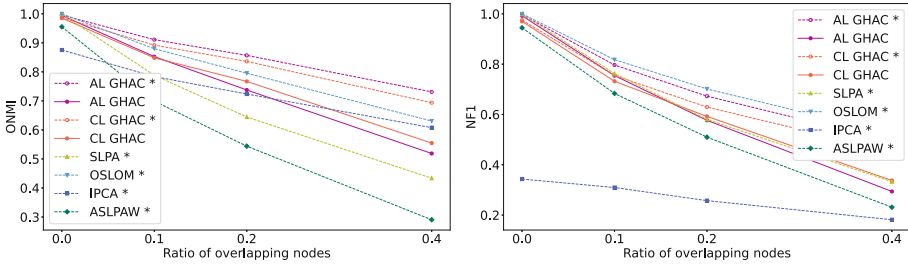


Fig. 1. Evaluations of overlapping node ratio impact to unweighted community detection performance for a subset A ($\mu = 0.1$).

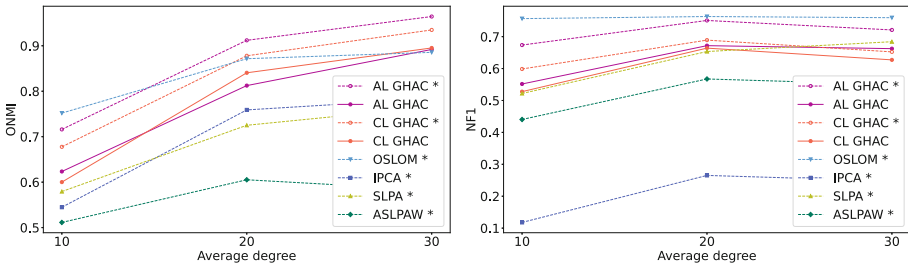


Fig. 2. Evaluations of network density impact to community detection performance. The evaluation was performed for subset B.

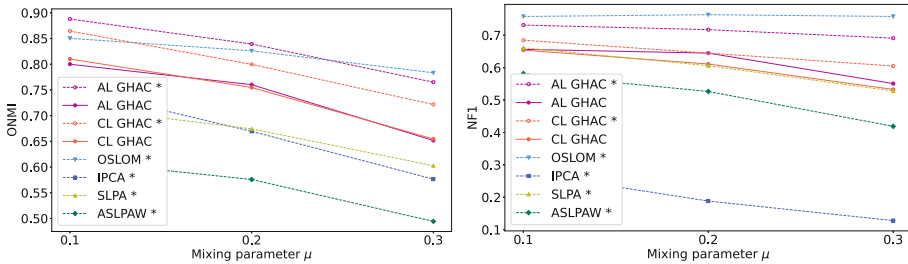


Fig. 3. Evaluations of mixing parameter μ impact to community detection performance. The evaluation was performed for subset B.

4.2 Sensitivity of Community Detection Methods to Edge Weight

The proposed methods are able to work with weighted networks. Therefore following experiments include a weighted version of networks. We are aware of existing versions of LFR benchmarks for the weighted graph generation. However, we noticed different behavior in intra-inter edge weight assignment when we used two different open-source implementations.

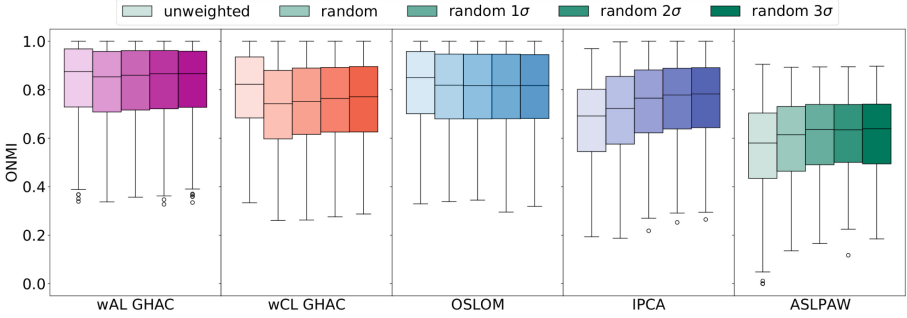


Fig. 4. Community detection sensitivity on the strengthening of intra-community edge weights for subset B.

For the sake of the simplicity of this experiment, we have decided to create random weights for already generated networks from the unweighted experiment. Therefore we are able to see the impact of weight information for the same graph. The procedure for weight generation consists of random sampling from a normal distribution $(\hat{\mu}, \sigma)$ and later strengthening of the weight inside of the community. The shift of intra-inter edges weight distribution was set to $1\sigma, 2\sigma, 3\sigma$. All weights were normalized into the interval $(1, 3)$ because of the use of d_{wCT} in the proposed dissimilarities.

We can notice the increasing ONMI value when the intra-community edge weights were increased from a random baseline in Fig. 4. The drop in score between unweighted and randomly weighted edges seems to be an effect of an applied procedure for weight generation. High sensitivity for generated weights is observed in the improvement of performance for methods IPCA and ASLPAW in contrast to OSLOM.

4.3 Weighted Synthetic Network

The following evaluation aims to explore the quality of detected network covers when the intra-community weights were strengthened by 3σ .

The relation between the number of overlapping nodes and evaluation scores ONMI and NF1 can be seen in Fig. 5. The comparison between proposed methods wAL GHAC and wCL GHAC to OSLOM yields in same observation as in comparison in unweighted experient, Fig. 1. The gap between wAL GHAC and wAL GHAC* is getting wider for a higher number of overlapping nodes. Different functions indicating the best cut in the hierarchy may help to achieve the upper performance bound. We have observed improvement in scores for IPCA

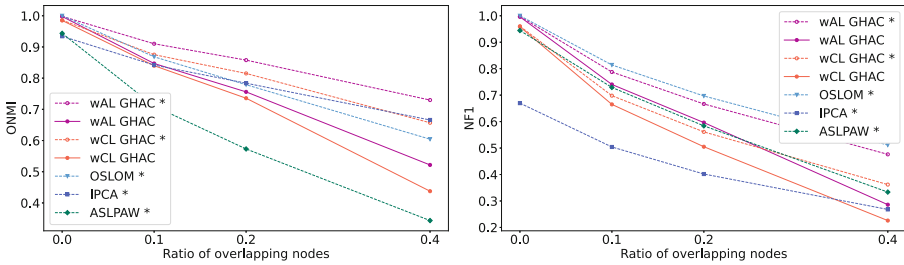


Fig. 5. Evaluations of overlapping node ratio impact to weighted community detection performance for a subset A ($\mu = 0.1$).

and ASLPAW, which is consistent with the observation from weight sensitivity boxplots in Fig. 4.

The relation of other specific network characteristics to community detection performance has been studied on the LFR benchmark. The study showed similar dependencies as in the unweighted experiment in the trend of observations gained by a number of overlapping node relation evaluations, observed in Fig. 5.

5 Conclusion

The hierarchical overlapping community detection methods were proposed for weighted networks. wAL GHAC and wCL GHAC achieved better scores for some configurations of the LFR benchmark at detecting overlapping communities than any of the other algorithms that have been tested. wAL GHAC achieved a better score in comparison to wCL GHAC. The dissimilarity between communities for the wAL GHAC method takes into account all distances between nodes from one community and the other, which better corresponds to the idea of a community where each node is close to the other. The hierarchy obtained by the proposed methods was not tested and compared with other methods but the result is naturally hierarchical and can be further used from a granular point of view on detected communities.

The proposed algorithms are sensitive to the weights of edges and the strengthening of intra-edge weights has a positive influence on the quality of detected communities. Modularity for identification of the specific level of cut in the hierarchy (as the quality of detected communities) is under discussion because the better value of ONMI and NF1 can be reached by proposed algorithms in different levels of the hierarchy. A better indication of the best cut than modularity would be beneficial.

Acknowledgment. This work was supported by SGS, VŠB – Technical University of Ostrava, Czech Republic, under the grant No. SP2023/12 “Parallel processing of Big Data X”.

References

1. Adamcsek, B., Palla, G., Farkas, I.J., Derényi, I., Vicsek, T.: CFinder: locating cliques and overlapping modules in biological networks. *Bioinformatics* **22**(8), 1021–1023 (2006)
2. Ahn, Y.Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. *Nature* **466**(7307), 761–764 (2010)
3. Altaf-Ul-Amin, M., Shinbo, Y., Mihara, K., Kurokawa, K., Kanaya, S.: Development and implementation of an algorithm for detection of protein complexes in large interaction networks. *BMC Bioinform.* **7**, 1–13 (2006)
4. Berahmand, K., Bouyer, A., Vasighi, M.: Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes. *IEEE Trans. Comput. Soc. Syst.* **5**(4), 1021–1033 (2018)
5. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech. Theor. Exp.* **2008**(10), P10008 (2008)
6. Bron, C., Kerbosch, J.: Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM* **16**(9), 575–577 (1973)
7. Brzozowski, L., Siudem, G., Gagolewski, M.: Community detection in complex networks via node similarity, graph representation learning, and hierarchical clustering (2023). arXiv preprint [arXiv:2303.12212](https://arxiv.org/abs/2303.12212)
8. Castrillo, E., León, E., Gómez, J.: Fast heuristic algorithm for multi-scale hierarchical community detection. In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pp. 982–989 (2017)
9. Clauset, A., Moore, C., Newman, M.E.: Hierarchical structure and the prediction of missing links in networks. *Nature* **453**(7191), 98–101 (2008)
10. Collins, L.M., Dent, C.W.: Omega: a general formulation of the rand index of cluster recovery suitable for non-disjoint solutions. *Multivar. Behav. Res.* **23**(2), 231–242 (1988)
11. Dao, V., Bothorel, C., Lenca, P.: Community structure: a comparative evaluation of community detection methods. *CoRR* [abs/1812.06598](https://arxiv.org/abs/1812.06598) (2018). <http://arxiv.org/abs/1812.06598>
12. El Ayeb, S., Hemery, B., Jeanne, F., Cherrier, E., Charrier, C.: Evaluation metrics for overlapping community detection. In: *2022 IEEE 47th Conference on Local Computer Networks (LCN)*, pp. 355–358. *IEEE* (2022)
13. Farkas, I., Ábel, D., Palla, G., Vicsek, T.: Weighted network modules. *New J. Phys.* **9**(6), 180 (2007)
14. Gregory, S.: Finding overlapping communities in networks by label propagation. *New J. Phys.* **12**(10), 103018 (2010)
15. Gupta, S.K., Singh, D.P., Choudhary, J.: A review of clique-based overlapping community detection algorithms. *Knowl. Inf. Syst.* **64**(8), 2023–2058 (2022)
16. Lancichinetti, A., Fortunato, S.: Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E* **80**(1), 016118 (2009)
17. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. *New J. Phys.* **11**(3), 033015 (2009)
18. Lancichinetti, A., Radicchi, F., Ramasco, J.J., Fortunato, S.: Finding statistically significant communities in networks. *PloS One* **6**(4), e18961 (2011)

19. Lázár, A., Abel, D., Vicsek, T.: Modularity measure of networks with overlapping communities. *EPL (Europhysics Letters)* **90**(1), 18001 (2010)
20. Lee, C., Reid, F., McDaid, A., Hurley, N.: Detecting highly overlapping community structure by greedy clique expansion (2010). arXiv preprint [arXiv:1002.1827](https://arxiv.org/abs/1002.1827)
21. Li, M., Chen, J.e., Wang, J.x., Hu, B., Chen, G.: Modifying the DPCLUS algorithm for identifying protein complexes based on new topological structures. *BMC Bioinform.* **9**(1), 1–16 (2008)
22. Li, T., et al.: Hierarchical community detection by recursive partitioning. *J. Am. Stat. Assoc.* pp. 1–18 (2020)
23. Li, X.L., Foo, C.S., Tan, S.H., Ng, S.K.: Interaction graph mining for protein complexes using local clique merging. *Genome Inform.* **16**(2), 260–269 (2005)
24. Lu, H., Sang, X., Zhao, Q., Lu, J.: Community detection algorithm based on non-negative matrix factorization and improved density peak clustering. *IEEE Access* **8**, 5749–5759 (2020)
25. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**(7043), 814–818 (2005)
26. Porter, M.A., Onnela, J.P., Mucha, P.J.: Communities in networks. *Not. AMS* **56**(9), 1082–1097 (2009)
27. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **76**(3), 036106 (2007)
28. Riolo, M.A., Cantwell, G.T., Reinert, G., Newman, M.E.J.: Efficient method for estimating the number of communities in a network. *CoRR* [abs/1706.02324](https://arxiv.org/abs/1706.02324) (2017). <http://arxiv.org/abs/1706.02324>
29. Rossetti, G.: graph benchmark handling community dynamics. *J. Complex Netw.* **5**(6), 893–912 (2017)
30. Rossetti, G., Milli, L., Cazabet, R.: CDLIB: a python library to extract, compare and evaluate communities from complex networks. *Appl. Netw. Sci.* **4**(1), 1–26 (2019)
31. Rossetti, G., Pappalardo, L., Rinzivillo, S.: A novel approach to evaluate community detection algorithms on ground truth. In: Cherifi, H., Gonçalves, B., Menezes, R., Sinatra, R. (eds.) *Complex Networks VII. SCI*, vol. 644, pp. 133–144. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30569-1_10
32. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.* **105**(4), 1118–1123 (2008)
33. Saoud, B., Moussaoui, A.: Node similarity and modularity for finding communities in networks. *Phys. A* **492**, 1958–1966 (2018)
34. Schaub, M.T., Li, J., Peel, L.: Hierarchical community structure in networks. *Phys. Rev. E* **107**(5), 054305 (2023)
35. Shen, H., Cheng, X., Cai, K., Hu, M.B.: Detect overlapping and hierarchical community structure in networks. *Phys. A* **388**(8), 1706–1712 (2009)
36. Shen, H.W., Cheng, X.Q., Guo, J.F.: Quantifying and identifying the overlapping community structure in networks. *J. Stat. Mech. Theor. Exp.* **2009**(07), P07042 (2009)
37. Simon, H.A.: The architecture of complexity. *Proc. Am. Philos. Soc.* **106**(6), 467–482 (1962)
38. Snášel, V., Dráždilová, P., Platoš, J.: Closed trail distance in a biconnected graph. *Plos One* **13**(8), e0202181 (2018)
39. Snášel, V., Dráždilová, P., Platoš, J.: Cliques are bricks for k-CT graphs. *Mathematics* **9**(11), 1160 (2021)

40. Suurballe, J.W., Tarjan, R.E.: A quick method for finding shortest pairs of disjoint paths. *Networks* **14**(2), 325–336 (1984)
41. Vieira, V.D.F., Xavier, C.R., Evsukoff, A.G.: A comparative study of overlapping community detection methods from the perspective of the structural properties. *Appl. Netw. Sci.* **5**, 51 (2020)
42. Wasserman, S., Faust, K.: *Social network analysis: Methods and applications*, vol. 8. Cambridge University Press (1994)
43. Xie, J., Szymanski, B.K., Liu, X.: SLPA: uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In: 2011 IEEE 11th International Conference on Data Mining Workshops, pp. 344–349. IEEE (2011)
44. Yang, J., Leskovec, J.: Overlapping community detection at scale: a nonnegative matrix factorization approach. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, pp. 587–596 (2013)
45. Zhang, X., Wang, C., Su, Y., Pan, L., Zhang, H.F.: A fast overlapping community detection algorithm based on weak cliques for large-scale networks. *IEEE Trans. Comput. Soc. Syst.* **4**(4), 218–230 (2017)
46. Zhang, Y., Levina, E., Zhu, J.: Detecting overlapping communities in networks using spectral methods. *SIAM J. Math. Data Sci.* **2**(2), 265–283 (2020)
47. Zhao, Y., Li, S., Wang, S.: Agglomerative clustering based on label propagation for detecting overlapping and hierarchical communities in complex networks. *Adv. Complex Syst.* **17**(06), 1450021 (2014)



Detecting Community Structures in Patients with Peripheral Nervous System Disorders

Morteza Hosseinioun^{1,2,3,4}(✉), Ali Mohammad Afshin Hemmatyar¹,
Saeid Ahmadifar², Hojjat Samiee³, and S. AmirAli Gh. Ghahramani⁴

¹ Sharif University of Technology, Azadi Ave., Tehran, Iran

morteza.hosseinioun@alum.sharif.edu, hemmatyar@sharif.edu

² Stavanger University, Kjell Arholms gate 41, 4021 Stavanger, Norway
s.ahmadifar@stud.uis.no

³ University of California, Davis, 1 Shields Ave, Davis, CA 95616, USA
info@samieechiropractic.com

⁴ Sharif University of Technology, International-Campus, Kish Island, Iran
ghahramani@pardis.sharif.edu

Abstract. Identifying sub-networks showing similar properties, referred to as community detection, is a challenging task in network analysis. This challenge becomes even more formidable in bipartite networks. The focus of this study is the patients with problems in their Peripheral Nerve System. To this aim, we engaged the assistance of spinal specialty clinics in the collection of necessary Data. We employ the bipartite network to represent the relationship between the patients and their symptoms and disorders. The resulting bipartite network showcases unequally sized sets of nodes, making community detection more challenging. The principal purpose of this study is to develop a new, practically relevant method for finding communities inside such networks. As such, we propose the Bi-MRComSim algorithm which applies different methods to transform the bipartite network to a unipartite one that can find meaningful communities between patients that coincide 85% of the time with diagnoses issued by physicians.

Keywords: Bipartite Network · Community Detection · Graph Projection · Peripheral Nervous System

1 Introduction

Our world is full of networks, representing various complex systems in different disciplines. Biological scientists, for example, seek to understand the associations between all known phenotypes and disease genes from a represented network of disorders and disease genes, whereas social scientists study the behavioral patterns of different groups of users from online social networks of acquaintanceships [20]. Many more instances may be found in politics, economics, marketing, computer science, transportation, and other fields [13, 18, 20, 21, 27].

However, how can we determine what constitutes dense communication or interactions between network components? The answer is a notion known as community detection. In network science, a community is defined as a set of nodes that are more likely to link to one another than to nodes from other communities [6]. Community detection is one of the fundamental concerns in social network analysis with the aim of identifying communities with uniform node and adherent structures. Although community detection has been widely explored, however, community detection of substantial networks with a high number of complexities remains a formidable obstacle [29]. For example, overlapping community detection is sensitive to the initialization of community centers, which may easily trap nodes in local optima and result in misclassification [31].

This research focuses on peripheral nerve system problems. We examined how individuals cope with their diseases and symptoms using data from a recognized neuroscience institute. In a bipartite network, patients and diseases or symptoms form two sets of nodes. The bipartite network has varied node sizes, making community detection more difficult. Three methods were used to project the resulting bipartite graph and its unique properties onto three unipartite graphs. The modified ComSim community detection algorithm [33], which we tailored to our study, received these unipartite graphs. The algorithm's accuracy was shown by comparing the found communities to the physician's suggestions.

2 Related Work

Numerous bipartite networks are seen in network theory. Many biological networks are naturally bipartite, such as the protein interaction network from yeast [17], the diseaseome network from Human Disease Network [5, 9], and the scientific collaboration network [26].

As these examples illustrate, in a bipartite graph the vertex set is partitioned into two disjoint subsets and each edge has two endpoints from two distinct vertex sets [4, 10]. Often, the purpose of studying these types of networks is to represent the interactions between members, such as economic linkages between countries, co-authorship, or actor-movie relationships, and to identify important communities.

In a node-based definition, a community is a set of nodes that connect more to each other than to other nodes of the network, based on the idea that they share the same resources or have similar properties. Many methods have been proposed based on this point of view, such as eigenvectors of matrices [23], modularity optimization [16], and clustering coefficient [34]. A well-known quality function that evaluates communities based on this idea is modularity [24].

On the other hand, in an edge-based definition, a community is a group of edges rather than nodes [1, 11]. The classification of edges into groups is based on the similarity of nodes connected to them. This definition is useful in dealing with overlapping communities, where each node inherits membership from all its incident edges and can belong to multiple communities. The choice of definition depends on the context and application requirements for a particular network.

Community detection in bipartite networks can be approached in different ways. One approach is to detect communities across the entire bipartite network. In another approach, the focus is on the disjoint sets of nodes, and the community structure within these sets is of importance. For this latter approach, the network will be projected onto a unipartite network, a procedure known as one-mode projection. This way, nodes belonging to the same set are linked if they have met some predefined conditions. Such approaches have been employed successfully for item recommendation based on customer-product and user-item bipartite networks [2, 3, 15, 30, 35]. One of the pioneer studies in this respect is the work presented by Freeman et al. [14] where during network projection, two nodes are connected if they have more comparable relations to one or more common nodes in the bipartite network. The Projection is frequently used to transform a bipartite network into a unipartite network suitable for further analysis.

Projecting a bipartite network into a unipartite counterpart often results in information loss. This is primarily due to the elimination of one set of vertices and the disregard for their connections. To address this limitation and minimize information loss, various studies of projection networks have advocated the use of a variety of quantitative functions and algorithms [7, 22]. Some strategies suggest decreasing the amount of information loss by incorporating weights on the edges of the resulting unipartite network. The most straightforward method of weighing is to count the number of neighbors that any two nodes in the projected unipartite network have in common in the original bipartite network [28]. Additionally, many bipartite modularity functions have been suggested, assuming that a community is a bipartite sub-graph made of nodes of both sets [8, 19, 32].

3 Problem Statement

The individuals who were the subject of this study and suffered from conditions affecting their peripheral nervous system were the primary focus of the research. Due to the existence of entities between patients and their disorders, our intention was to use a bipartite network to illustrate the connection that exists between patients and the disorders that they experience. In order to fulfill this objective, we engaged the assistance of a spinal specialist clinic in the process of collecting the necessary data about patients. As stated previously, these patient-disorder networks are bipartite, which results in sets of nodes that are not of equal size in both types A and B. Also, the main objective of this study was to develop a method that might be used for identifying community values inside these types of networks. An algorithm was then built to assess the unique aspects of this dataset. This will allow for the detection of correlations between patients and the disorders that they suffer from. The method, which goes by the name Bi-MRComSim, makes use of three different approaches, each of which is an example of a possible approach for projecting a bipartite network into a unipartite graph. In addition, after projecting the unipartite networks, we

sent them through a community detection algorithm in order to determine the community values derived from each method.

4 Dataset Description

Our dataset comprises information on 311 patients who sought treatment at a specialized spinal clinic. For each patient, we documented issues pertaining to their cervical and lumbar regions, as well as general symptoms. As a result, each patient was associated with a set of potential disorders. We transformed this data into a bipartite graph, featuring two types of nodes: Type A nodes representing the patients and Type B nodes representing the disorders they suffered from. We established connections between each patient and the disorders they were experiencing. Recognizing that disorders related to different parts of the spine hold varying levels of importance, we decided to assign distinct weights to these disorders based on input from the spinal clinic’s specialists. Specifically, disorders related to cervical regions were assigned a weight with a value of 6, lumbar region disorders were given a weight of 3, and other general disorders received a weight of 1. Consequently, we created a weighted bipartite graph in which the number of Type A nodes far exceeded that of Type B nodes. The resulting bipartite graph, depicted in Fig. 1, was used as the foundation for our analysis.

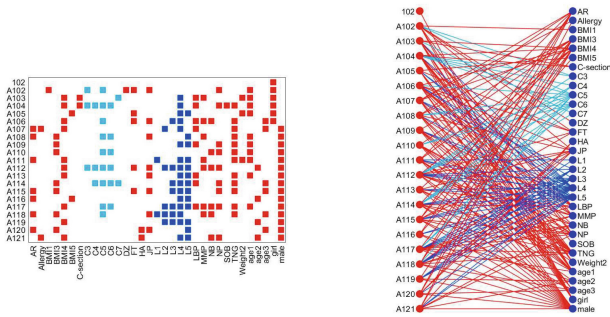


Fig. 1. The weighted bipartite graph depicts three groups with three colors: red for general disorder symptoms, pale blue for cervical, and dark blue for lumbar.

The main network characteristics of the resulting bipartite network have been reported in Table 1.

Physicians at the spinal clinic had previously clustered the patients into six distinct communities based on their disorders, symptoms, and explicit diagnoses. These communities serve as our reference or “ground-truth” communities for assessing the performance of our proposed community detection algorithm.

Table 1. Properties of the Resulting Bipartite Network.

Network Overview	
Number of Type A nodes	311
Number of Type B Nodes	51
Number of Edges	3996
Avg. Clustering Coefficient	0
Network Diameter	1
Connected Components	1

5 Proposed Method

To detect the communities between patients in the bipartite graph described in 3, we propose the Bi-MRComSim algorithm. The Bi-MRComSim encompasses three main phases as follows:

- The projection phase: The bipartite graph will be transformed into a unipartite graph of patients
- Weight assignment phase: In this phase, for each edge in the unipartite network, a weight will be created.
- The community detection phase: inspired by the ComSim algorithm, the communities among the patients in the projected unipartite graph will be detected

5.1 The Projection Phase

Let $G(A, B, E)$ be the bipartite graph with $|A(G)| = n_1$ be patients vertex set, $|B(G)| = n_2$ be the disorders vertex set, and $|E(G)| = m$ be the total number of links in the bipartite network, Now projection of the bipartite graph G for the vertex set A with respect to the vertex set B is to construct a unipartite or one mode network $G'(U, E')$ where $V(G) = A$ and $(a_i a_j) \in E(G')$ if $N(A_i) \cap N(A_j) \neq \phi$, where $N(A_i)$ is the set of neighbors linked to node A_i from the disorders vertex set (vertex set B). In other words, if any two patients share at least one common disorder, there will be a link between them in the resulting unipartite projected network. The procedure of this projection is shown in Algorithm 1.

5.2 Weight Assignment Phase

After the projection phase, the next step is to assign weights to the edges of the resulting unipartite network. To this aim, three weight assignment procedures have been proposed. In the following, we elaborate on these procedures.

$\frac{Meet}{Union}$ Function Let e_{ij} be an edge between nodes A_i and A_j in the projected unipartite network. The weight assigned to this edge will be calculated in the following

Data: Bi-adjacency Matrix (B) of the Bipartite Network.

Result: Adjacency Matrix (A) of the Projected Network.

```

begin
     $n_1 \leftarrow B.NoOfRows()$ 
     $n_2 \leftarrow B.NoOfColumns()$ 
    for  $i \in n_1$  do
        for  $j = i + 1 \in n_1$  do
            for  $k \in n_2$  do
                if  $B[i][k] == 1$  and  $B[j][k] == 1$  then
                     $A[i][j] = 1$ 
                    break
                else
                     $A[i][j] = 0$ 
                end
            end
        end
    end
end
    
```

Algorithm 1: Algorithm for Computing Projected Network

$$weight = \frac{\sum_{k \in (N(A_i) \cap N(A_j))} Weight(k)}{\sum_{k \in (N(A_i) \cup N(A_j))} Weight(k)} \quad (1)$$

where $N(A_i)$ is the set of nodes from the disorders vertex set (vertex type B) that have a link with node A_i in the original bipartite graph and $Weight(k)$ denotes the weight associated with the disorder as what we explained in the description of the dataset. This procedure is shown in Algorithm 2

Data: Unweighted Projected Unipartite Network, Original Weighted Bipartite Network

Result: Weighted Unipartite Network with $\frac{Meet}{Union}$ weights initialization;

```

while there are any two nodes in the unweighted projected unipartite network do
    Meet = Find  $\sum_{k \in (N(A_i) \cap N(A_j))} Weight(k)$ 
    Union = Find  $\sum_{k \in (N(A_i) \cup N(A_j))} Weight(k)$ 
    Link Weight =  $\frac{Meet}{Union}$ 
    Assign link Weight to edge
end
    
```

Algorithm 2: $\frac{Meet}{Union}$ function

$\frac{Meet}{Union} \times$ **Random Walk Similarity Function.** As the second strategy for calculating weights, we used the $\frac{Meet}{Union} \times RandomWalk$ technique. The random walk approach examined in this research is based on the Page rank algorithm

[25]. We use this approach to determine the degree of similarity between every pair of type A nodes based on the characteristics found on the other side of the weighted bipartite graph. The random walk procedure can be explained as follows. First, we create a counter for each type A node in the bipartite graph and we initialize it to zero. For each type A node we run the following procedure a large number of times:

- 1: We randomly select one of the type B neighbors of the node and move to it on the other side of the bipartite network. This random selection is proportional to the weight considered for that particular connection.
- 2: On the other side of the network, we randomly select one of its neighbors (type A nodes) and move to it. Moreover, we increment the counter associated with this type A node for each visit.
- 3: We repeat steps 1 and 2

After a large number of iterations, we focus on the counter of type A nodes. To normalize the counter values, we can apply the $\frac{\text{number of visits}}{\text{number of iterations}}$. Those nodes with higher normalized counter values are more similar to the target node. Now, the weight of edge between nodes A_i and A_j can be computed as:

$$\text{weight} = \frac{\sum_{k \in (N(A_i) \cap N(A_j))} \text{Weight}(k)}{\sum_{k \in (N(A_i) \cup N(A_j))} \text{Weight}(k)} \times \text{RandomWalkScore}(A_i, A_j) \quad (2)$$

Threshold Usage Function. In the third strategy, we try to combine the two weighting procedures proposed in previous sections. Let e_{ij} be an edge in the projected unipartite network connecting nodes A_i and A_j . We compute a similarity score between A_i and A_j based on the random walk score (i.e., $\text{RandomWalkScore}(A_i, A_j)$). If this similarity score is greater than a predefined threshold, the weight of the edge would be computed based on the $\frac{\text{Meet}}{\text{Union}}$ weighting strategy. Otherwise, this link will be removed from the unipartite network. This procedure has been shown in the Algorithm 3.

5.3 The Community Detection Phase

In this phase and inspired by the ComSim algorithm, we find the community among type A nodes (the patients) in the unipartite projected network. The community detection is composed of two steps.

In the initial phase, we initiate the identification of core communities. To do this, we commence by identifying a type A node with the highest degree, which we refer to as the “target node.” Subsequently, we seek out a neighbor of the target node with the greatest similarity, where similarity is defined by the weight associated with the edges connecting the target node to its neighbors. The newly selected node now becomes the target node, and we repeat this process until we revisit a previously explored node, indicating the formation of a cycle. At this point, all the nodes involved in the cycle are designated as the constituents of a core community. Following this, we proceed to select the next type A node with

Data: Unweighted Projected Unipartite Network, Original Weighted Bipartite Network
Result: Weighted Unipartite Network with Threshold Usage initialization;
 Look at the link between every two nodes (A_i, A_j) of the graph
for $i, j \in All2NodesInGraph$ **do**
 if $RandomWalkScore(A_i, A_j) > threshold$ **then**
 $LinkWeight = \frac{\sum_{k \in (N(A_i) \cap N(A_j))} Weight(k)}{\sum_{k \in (N(A_i) \cup N(A_j))} Weight(k)}$
 else
 Delete Link
 end
end

Algorithm 3: Threshold function

the highest degree and repeat the aforementioned procedure. Once this process is completed, we will have successfully partitioned the nodes that constitute the core communities. This procedure is detailed in Algorithm 4.

In the second step, our objective is to assign the remaining nodes from the first step to the most suitable core communities. To achieve this, we identify, for each remaining node, those core communities that have at least one connection with it. Subsequently, we allocate the remaining node to the core community which maximizes the total similarity score between the remaining node and all the nodes within that core community. This procedure is illustrated in Algorithm 5

6 Experiments and Results

In this section, we compare Bi-MRComSim to physician diagnosis. The implementation of Bi-MRComSim uses BiMAT [12], a ©MATLAB library for bipartite network analysis and visualization. The application development and running PC has 16 GB RAM with ©MATLAB R2022a. The suggested Bi-MRComSim approach is tested by comparing its findings to the DofD. In order to test randomness, the algorithm iterations are limited to 1,000,000.

It is essential to note that our model, Bi-MRComSim, may generate small communities, typically comprising only two members, which may not align directly with the physician’s reported communities. To address this issue, we evaluate the similarity between these small communities and the larger, core communities. This similarity is defined as the sum of link weights connecting members of a small community to members of a larger community within the weighted unipartite network projection. Subsequently, we identify the core community with the highest similarity and incorporate the members of the small community into it.

To compute accuracy formally, we establish two distinct groups of communities: the “target communities” designated by physicians and the “predicted

Data: a bipartite graph $\beta = (\text{type A, type B, Edge, a similarity function } Similarity)$ and the weighted projected unipartite graph

Result: return a partition P of type A nodes and a set K of remaining nodes (for the second step)

```

P:= $\phi$  // the partition set
T:= type A // the set of nodes to be considered
 $\chi$ := PickMaxDegreeAndRemove(T) // the highest degree type A node
V:= $\phi$  // set of nodes currently considered
K:= $\phi$  // set of remaining nodes
initialization;
while  $T \neq \emptyset$  do
  /* finds a neighbor  $y$  of  $\chi$  maximizing  $Similarity(\chi, y)$ 
  if  $y \in V$  then
    C := cycle(V, y,  $\chi$ ) // extract the detected cycle from y to x in V
    P.add(C)
    K := K  $\cup$  (V - C) // stores nodes not in the cycle C
    V :=  $\phi$ 
     $\chi$  := PickMaxDegreeAndRemove(T)
  else
    if  $y \in V$  then
      V := V  $\cup$  y
      x := y
      T := T - y
    else
      /* y is already part of an element of P, visited nodes are stored */
      K := K  $\cup$  V
      V :=  $\phi$ 
       $\chi$  := PickMaxDegreeAndRemove(T)
    end
  end
end
return P and K

```

Algorithm 4: Community Detection First Step

communities” generated by Bi-MRComSim after excluding the small communities. Here’s the formal accuracy calculation:

- We compute the intersection between each predicted community and all of the target communities. The intersection represents the common members shared between a predicted community and the target communities.
- We identify the highest intersection value obtained for each predicted community concerning all target communities.
- To determine the accuracy for a specific predicted community, we divide the highest intersection value by the size of the predicted community.
- To obtain the overall accuracy, we calculate the mean of the accuracies for all predicted communities.

For instance, if a predicted community contains 50 members, and its highest intersection with the target communities is 30 members, then the accuracy for

Data: a bipartite graph $\beta = (\text{type A, type B, Edge})$; a partition P ; a set K of remaining nodes (from **first step**), a similarity function *Similarity*, weighted projected unipartite graph

Result: return a partition P' of type A nodes and unsatisfied nodes R

$R := \phi$ // Remaining nodes

$P' := P$

initialization;

while $X \in K$ **do**

$P_X := \text{CommonNeighbor}(\chi, P)$ // Find all neighbor communities of x

if $P_X := \phi$ **then**

$R := R \cup \{X\}$

else

$C := \text{argmax}_{C_x \in P_x} \sum_{y \in C_x} \text{Similarity}(x, y)$

 Add X into the partition C of P'

end

end

return P' and R

Algorithm 5: Community Detection Second phase

Table 2. Results

Methods	Results
$\frac{Meet}{Union}$	0.8590
Random Walk	0.8487
Threshold	0.7331

this predicted community would be $30/50 = 0.6$. The total accuracy is then computed as the average of these individual community accuracies.

This formal accuracy assessment method allows us to quantitatively evaluate the agreement between the communities detected by Bi-MRComSim and those identified by medical professionals, providing a comprehensive and rigorous evaluation of our model's performance.

6.1 Results

In this part, we compare the performance of Bi-MRComSim with standard DoFD. For final measurement, the program is executed five times and the average number of those five replicates was considered as the final number.

Table 2 demonstrates that the $\frac{Meet}{Union}$ approach has the highest accuracy compared to all other methods. This is due to the fact that increasing the connectness between patients' neighbors has a more robust rationale than the other two methods. Therefore, the $\frac{Meet}{Union}$ technique more precisely covers the severity of disorders amongst patients. Therefore, the community function has a superior output for examining the relationship between the outcome of our algorithm and the doctor's diagnosis compared to other functions.

As seen in Fig. 2a, the analysis of each of the three methodologies reveals a consistent pattern in achieving community structures. The testing results show that when $\frac{Meet}{Union}$ is used, Bi-MRComSim is superior to almost 86%. Practically speaking, the $\frac{Meet}{Union}$ method is the best option since it finds roughly 86% of the connections between the DofD clusters and the algorithmically derived community structures. Furthermore, the value of $\alpha = 0.1259$ represents the variation error seen in all methodologies in relation to DofD. Accordingly, both our best and least accurate models' errors in identifying community structures are close to $\alpha \cong 13\%$.

As seen in Fig. 2b, the $\frac{Meet}{Union}$ method serves as the primary method of enlarging the number of communities. The findings illustrate a straight correlation between the number of communities and the number of nodes, which is essential in the $\frac{Meet}{Union}$ methodology for identifying precise community components.

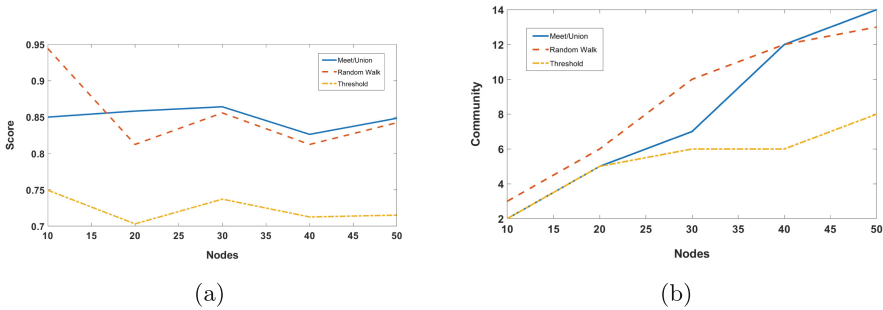


Fig. 2. (a) is the deviations of three methods (b) is the number of Communities

In this study, we tackled the challenge of detecting community structures in patients with peripheral nervous system (PNS) disorders using a bipartite network representation. Our dataset included 311 patients and their associated symptoms and disorders, resulting in an unequal distribution of nodes in the bipartite network. To address this, we proposed the Bi-MRComSim algorithm, which consists of three key phases: projection, weight assignment, and community detection. In the projection phase, we transformed the bipartite network into a unipartite one, focusing on patients. We then assigned weights to the edges in various ways, such as using the $\frac{Meet}{Union}$ function, random walk similarity, or a threshold-based approach. Finally, we applied a community detection algorithm, akin to the ComSim algorithm, to find communities among patients. We evaluated our method by comparing the detected communities to physician diagnoses and found that the $\frac{Meet}{Union}$ method achieved the highest accuracy, coinciding with diagnoses approximately 86% of the time. Overall, our study offers a practical approach to detecting community structures in PNS disorder patients, which can aid in understanding and managing these conditions.

References

1. Ahn, Y.Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. *Nature* **466**(7307), 761 (2010)
2. An, Y.H., Dong, Q., Sun, C.J., Nie, D.C., Fu, Y.: Diffusion-like recommendation with enhanced similarity of objects. *Phys. A* **461**, 708–715 (2016)
3. Bakshy, E., Rosenn, I., Marlow, C., Adamic, L.: The role of social networks in information diffusion. In: *Proceedings of the 21st International Conference on World Wide Web*, pp. 519–528. ACM (2012)
4. Banerjee, S., Jenamani, M., Pratihari, D.K.: Properties of a projected network of a bipartite network. In: *Communication and Signal Processing (ICCSP), 2017 International Conference*, pp. 0143–0147. IEEE (2017)
5. Barabási, A.L., Gulbahce, N., Loscalzo, J.: Network medicine: a network-based approach to human disease. *Nat. Rev. Genet.* **12**(1), 56 (2011)
6. Barabási, A.L., Pósfai M.: *Network science*. Cambridge University Press, Cambridge (2016). <http://barabasi.com/networksciencebook/>
7. Barber, M.J.: Modularity and community detection in bipartite networks. *Phys. Rev. E* **76**(6), 066102 (2007)
8. Barber, M.J., Faria, M., Streit, L., Strogan, O.: Searching for communities in bipartite networks. In: *AIP Conference Proceedings*, vol. 1021, pp. 171–182. AIP (2008)
9. Börner, K.: *Atlas of science*, vol. 2. Citeseer (2010)
10. Descartes, B., et al.: Ja Bondy and USR Murty, graph theory with applications. *Bull. Am. Math. Soc.* **83**(3), 313–315 (1977)
11. Evans, T., Lambiotte, R.: Line graphs, link partitions, and overlapping communities. *Phys. Rev. E* **80**(1), 016105 (2009)
12. Flores, C.O., Poisot, T., Valverde, S., Weitz, J.S.: BiMat: a MATLAB (r) package to facilitate the analysis and visualization of bipartite networks (2014). arXiv preprint [arXiv:1406.6732](https://arxiv.org/abs/1406.6732)
13. Fortunato, S., Hric, D.: Community detection in networks: a user guide. *Phys. Rep.* **659**, 1–44 (2016)
14. Freeman, L.C.: Finding social groups: a meta-analysis of the southern women data. In: *Dynamic Social Network Modeling and Analysis*. The National Academies, pp. 39–97. Press (2003)
15. Guille, A., Hacid, H., Favre, C., Zighed, D.A.: Information diffusion in online social networks: a survey. *ACM SIGMOD Rec.* **42**(2), 17–28 (2013)
16. Guimerà, R., Sales-Pardo, M., Amaral, L.A.N.: Module identification in bipartite and directed networks. *Phys. Rev. E* **76**(3), 036102 (2007)
17. Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., Barabási, A.L.: The large-scale organization of metabolic networks. *Nature* **407**(6804), 651 (2000)
18. Kenett, D., Havlin, S.: Network science: a useful tool in economics and finance. *Mind Soc.* **14**, 155–167 (2015). <https://doi.org/10.1007/s11299-015-0167-y>
19. Li, Z., Wang, R.S., Zhang, S., Zhang, X.S.: Quantitative function and algorithm for community detection in bipartite networks. *Inf. Sci.* **367**, 874–889 (2016)
20. Mej, N.: *Networks: an introduction* (2010)
21. Mozzoni, S., Francesco, M.D., Maternini, G., Barabino, B.: Transfer’ monitoring in bus transit services by automatic vehicle location data. *Transp. Res. Procedia* **60**, 402–409 (2022). <https://doi.org/10.1016/j.trpro.2021.12.052>, <https://www.sciencedirect.com/science/article/pii/S2352146521009534>. New scenarios for safe mobility in urban areas *Proceedings of the XXV International Conference Living and Walking in Cities (LWC 2021)*, September 9–10, 2021, Brescia, Italy

22. Murata, T., Ikeya, T.: A new modularity for detecting one-to-many correspondence of communities in bipartite networks. *Adv. Complex Syst.* **13**(01), 19–31 (2010)
23. Newman, M.E.: Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **74**(3), 036104 (2006)
24. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E* **69**(2), 026113 (2004)
25. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Tech. rep, Stanford InfoLab (1999)
26. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**(7043), 814 (2005)
27. Patty, J.W., Penn, E.M.: 147Network Theory and Political Science. In: *The Oxford Handbook of Political Networks*. Oxford University Press (2017). <https://doi.org/10.1093/oxfordhb/9780190228217.013.12>
28. Ramasco, J.J., Morris, S.A.: Social inertia in collaboration networks. *Phys. Rev. E* **73**(1), 016122 (2006)
29. Rostami, M., Oussalah, M.: A novel attributed community detection by integration of feature weighting and node centrality. *Online Soc. Netw. Media* **30**, 100219 (2022). <https://doi.org/10.1016/j.osnem.2022.100219>. <https://www.sciencedirect.com/science/article/pii/S2468696422000234>
30. Shang, M.S., Fu, Y., Chen, D.B.: Personal recommendation using weighted bipartite graph projection. In: *Apperceiving Computing and Intelligence Analysis, 2008. ICACIA 2008. International Conference on*, pp. 198–202. IEEE (2008)
31. Shang, R., Zhao, K., Zhang, W., Feng, J., Li, Y., Jiao, L.: Evolutionary multi objective overlapping community detection based on similarity matrix and node correction. *Appl. Soft Comput.* **127**, 109397 (2022). <https://doi.org/10.1016/j.asoc.2022.109397>, <https://www.sciencedirect.com/science/article/pii/S1568494622005361>
32. Suzuki, K., Wakita, K.: Extracting multi-facet community structure from bipartite networks. In: *2009 International Conference on Computational Science and Engineering*, pp. 312–319. IEEE (2009)
33. Tackx, R., Tarissan, F., Guillaume, J.-L.: COMSIM: a bipartite community detection algorithm using cycle and node’s similarity. In: Cherifi, C., Cherifi, H., Karsai, M., Musolesi, M. (eds.) *COMPLEX NETWORKS 2017 2017*. SCI, vol. 689, pp. 278–289. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-72150-7_23
34. Zhang, P., Wang, J., Li, X., Li, M., Di, Z., Fan, Y.: Clustering coefficient and community structure of bipartite networks. *Phys. A* **387**(27), 6869–6875 (2008)
35. Zhou, T., Ren, J., Medo, M., Zhang, Y.C.: Bipartite network projection and personal recommendation. *Phys. Rev. E* **76**(4), 046115 (2007)



Community Detection in Feature-Rich Networks Using Gradient Descent Approach

Soroosh Shalileh^{1,2(✉)} and Boris Mirkin^{3,4}

¹ Center for Language and Brain, HSE University, Myasnitskaya Ulitsa, 20, Moscow 101000, Russia

² Vision Modelling Lab, HSE University, Myasnitskaya Ulitsa, 20, Moscow 101000, Russia

sshalileh@hse.ru

³ Department of Data Analysis and Artificial Intelligence, HSE University, Pokrovsky Boulevard, 11, 101000 Moscow, Russia

bmirkin@hse.ru

⁴ Department of Computer Science and Information Systems, Birkbeck University of London, Malet Street, London WC1E 7HX, UK

Abstract. The gradient descent has proven to be an effective optimization strategy. The current research proposes a novel clustering methodology using this strategy to recover communities in feature-rich networks. Our adoption of this strategy did not lead to promising results, and thus to improve them, we propose a special “refinement” mechanism, which culls out potentially misleading objects during the optimization. We validated and compared our proposed methods with three state-of-the-art algorithms over four real-world and 160 synthetic data sets. Our results proved that our proposed method is valid and in the majority of cases has a significant edge over the competitors.

Keywords: feature-rich networks · community detection · steepest descent optimization · gradient descent approach

1 Introduction: Background and Modification

Community detection using gradient descent (GD) in various modifications is becoming popular because it brings forth such advantages as a continuous representation of difficult combinatorial problems, reasonable non-linearity formulations, and universality. A recent review of techniques used can be found in [1]. The developments concentrate, first of all, on using GD for training networks participating in optimization of community detection algorithms, like it is done in [1, 9]. The former involves the modularity function, the latter the spectral approach criterion. The other direction is based on reformulation of the community detection problem as a relaxed continuous model involving “soft” label assignments functions. A further on elaboration of the model may go either by using neural network for a specific GD based method of optimization (see, for example, [9, 11, 24]) or without any involvement of neural networks, by formulating a

specific relaxation of the problem and a problem-specific method for its solution. For example, in [5] a quadratic programming relaxation of the node matching problem is developed, whereas authors of [4] formulate a specific probabilistic structure with a corresponding criterion. Both use GD for solving corresponding problems. Authors of [8] introduce elements of a GD framework for modularity optimization using what they call a vector-label propagation algorithm, in which a node is associated with a vector of continuous community labels (instead of one label, as usual).

In contrast, the authors of current research utilize a straightforward data recovery community detection model such as described in our work [20]. We apply GD to the least-squares data-recovery criterion without relaxing its combinatorial structure, like it was done by Shalileh S. in [19]. Unfortunately, in real-world computations, the GD algorithms for the least-squares criterion, even in improved versions such as that by Kingma and Ba [12], may encounter "bad sequences" of objects and converge to points which are far from optimal. To tackle this issue, we introduce a special "refinement" mechanism, which culls out potentially misleading objects. Our goal is to show that our refined gradient descent method indeed leads to improvement in the quality of results.

2 Methodology

2.1 Problem Formulation

Let $D = \{X, A\}$ represent a feature-rich network over a nodes' set I with the cardinality N ; where $X \in \mathbb{R}^{N \times V}$ and $A \in \mathbb{R}^{N \times N}$, represent the features and the adjacency matrices, respectively. We aim to partition D into K crisp clusters using the feature and network information simultaneously such that the within cluster entities are as interconnected as possible and share similar features.

To achieve our objective, we associate each cluster, \mathbf{s}_k , with the centroid in the feature space \mathbf{c}_k , and the centroid in the network space $\boldsymbol{\lambda}_k$, forming the set of clusters, $S = \{\mathbf{s}_k\}_{k=1}^K$, the set of centroids in feature space, $C = \{\mathbf{c}_k\}_{k=1}^K$, and the set of centroids in network space, $A = \{\boldsymbol{\lambda}_k\}_{k=1}^K$ respectively. With this notation we can define a generic clustering objective function as:

$$J(D, C, A) = \sum_{i=1}^N \sum_{k=1}^K \rho f(\mathbf{c}_k, \mathbf{x}_i) + \xi h(\boldsymbol{\lambda}_k, \mathbf{a}_i). \quad (1)$$

where $f(\cdot) : \mathbb{R}^V \rightarrow \mathbb{R}$ and $h(\cdot) : \mathbb{R}^N \rightarrow \mathbb{R}$ represent a distance functions of the choice that will be applied to measure the distance between the "i-th" data points, in the feature and network spaces, and the corresponding centroids. In this work, being differentiable is the only limit we have for these distance functions. And to adjust the trade between the two data sources during the clustering procedure we define $\rho, \xi \in [0, 1]$. In the remainder of the current study we fixed them to unity.

Optimizing the objective function in Eqn. (1), even using a straightforward distance, like Euclidean distance, is computationally expensive and cannot be solved exactly in a reasonable time. In this work, inspired by the triumph

of the gradient descent approach and automatic differentiation in supervised learning tasks, and following the research direction started [19], we proposed a novel methodology to recover clusters in feature-rich networks using the gradient descent approach, and automatic differentiation. Since the gradient is the direction of the steepest descent and is the underlying concept of our proposed methodology, we name our proposed method the steepest descent clustering (SDC).

2.2 Proposed Clustering Methods

SDC method consists of three components: (i) cluster assignment criterion, (ii) cluster update rule(s), and (iii) convergence condition. Before explaining them, we update our notation by adding subscript t to represent the concept of iterations. Concretely, we denote the set of centroids in the feature space and network space at iteration t with $C^{(t)} = \{\mathbf{c}_k^{(t)}\}_{k=1}^K$ and $\Lambda^{(t)} = \{\boldsymbol{\lambda}_k^{(t)}\}_{k=1}^K$ respectively. Similarly, we denote the set of detected clusters at iteration t with $S^{(t)} = \{\mathbf{s}_k^{(t)}\}_{k=1}^K$.

The Eqn. (2) represents the first component of SDC, i.e., the cluster assignment criterion:

$$\underset{k}{\operatorname{argmin}} f(\mathbf{x}_i, \mathbf{c}_k^{(t)}) + h(\mathbf{a}_i, \boldsymbol{\lambda}_k^{(t)}) < f(\mathbf{x}_i, \mathbf{c}_j^{(t)}) + h(\mathbf{a}_i, \boldsymbol{\lambda}_j^{(t)}), \forall j \neq k. \quad (2)$$

that is, at iteration t the i -th node will be assigned to the cluster k for which the total sum of the distances, between the i -th data points and the corresponding centroids, in both spaces, is minimized.

For brevity, we only concentrate on explaining the centroid update rules in the features space. Similar equations can be obtained for the corresponding centroids in the network space.

The second component of our proposed method, i.e., the update rule, in its vanilla form (VSDC), for the feature-space centroids, is explained in Eqn. (3):

$$\mathbf{c}_k^{(t+1)} = \mathbf{c}_k^{(t)} - \alpha \nabla_{\mathbf{c}_k^{(t)}} f(\mathbf{x}_i, \mathbf{c}_k^{(t)}) \quad (3)$$

where α represents the step size, and $\nabla_{\mathbf{c}_k^{(t)}}$ is the gradient of the distance function, f , w.r.t the k -th feature centroid at iteration t evaluated with the feature data point \mathbf{x}_i . It ought to be added that our preliminary experiments, aligned with the results of [23], proved that the on-line update rule is significantly more efficient than the batch version of the gradient descent algorithm; therefore, we update the centroids in an on-line manner, that is the each batch consists of only one data point.

It is well-known that the vanilla update rule is prone to slow convergence, especially at nearly flat surfaces [13]. To tackle this issue, the Nesterov accelerated momentum [18] was proposed. However, the vanilla update rule and the Nesterov accelerated momentum rule update the components of the centroids with the constant step size; that leads to dominantly, and not necessarily desirably, influencing the update direction by those components with constantly high

gradients. The adaptive gradient optimization methods such as [12, 25] have been proposed to address this issue. In the current study, we utilize the adaptive moment estimation method [12], also called ADAM, and propose the ASDC update rule to increase the influence of components with less frequent high gradients during the centroid updates.

To describe the ASDC update rule, let us denote the gradient vector in the feature space at iteration t with $\mathbf{g}_k^{(t)} = \nabla_{\mathbf{c}_k^{(t)}} f(\mathbf{x}_i, \mathbf{c}_k^{(t)})$. And let $\mathbf{g}_k^{(t)} \odot \mathbf{g}_k^{(t)}$ be element-wise product of this gradient vector by itself. Also, let us denote the first-moment (the mean) and second-moment (the uncentered variance) vectors in the feature space with $\mathbf{m}^{(t)}$ and $\mathbf{n}^{(t)}$, respectively. The Eqn. (4) represents the four steps of the ASDC in the feature and network space, respectively:

$$\mathbf{m}^{(t+1)} = \beta_1 \mathbf{m}^{(t)} + (1 - \beta_1) \mathbf{g}_k^{(t)} \quad (4a)$$

$$\mathbf{n}^{(t+1)} = \beta_2 \mathbf{n}^{(t)} + (1 - \beta_2) (\mathbf{g}_k^{(t)} \odot \mathbf{g}_k^{(t)}) \quad (4b)$$

$$\hat{\mathbf{m}}^{(t+1)} = \mathbf{m}^{t+1} / (1 - \beta_1^t) \quad (4c)$$

$$\hat{\mathbf{n}}^{(t+1)} = \mathbf{n}^{t+1} / (1 - \beta_2^t) \quad (4d)$$

$$\mathbf{c}_k^{(t+1)} = \mathbf{c}_k^{(t)} - \alpha \hat{\mathbf{m}}^{(t+1)} / (\sqrt{\hat{\mathbf{n}}^{(t+1)}} + \epsilon), \quad (4e)$$

where $\mathbf{m}^{(t)}$, $\mathbf{n}^{(t)}$, $\mathbf{r}^{(t)}$, and $\mathbf{q}^{(t)}$ are initialized with zero, and will be calculated using the exponential moving averages of estimates of the gradient and its square, and $\epsilon = 1 \times 10^{-8}$. More precisely, the first and the second moments of feature gradient vector can be formulated as:

$$\mathbf{m}^{(t+1)} = (1 - \beta_1) \sum_{j=0}^t \beta_1^{t-j} \mathbf{g}_k^{(j)}, \quad (5)$$

and

$$\mathbf{n}^{(t+1)} = (1 - \beta_2) \sum_{j=0}^t \beta_2^{t-j} (\mathbf{g}_k^{(j)} \odot \mathbf{g}_k^{(j)}). \quad (6)$$

where $\beta_1, \beta_2 \in [0, 1)$ are the corresponding decay weights. Since the adjacency matrix is usually sparser than the feature matrix, the behavior of gradients in these two spaces are different, and we expect noisier behaviors in the network space. Moreover, we applied this update rule in clustering, thus, we treated the four decay weights, two corresponding to the features and two corresponding to the network, as hyperparameters and tuned them separately in Sect. 4.

As an additional caution, to correct bias due to zero initializations of moments and to prevent taking a large step size at the beginning of the optimization, we divide them with $(1 - \beta_1^t)$ and $(1 - \beta_2^t)$, in (4c) and (4d) respectively. It ought to be mentioned that the superscript t denotes the exponentiation power, while (t) denotes the iteration number. The network gradient moments can be calculated similarly and for brevity, we avoid discussing them here.

Theoretically various convergence conditions can be adopted for the last constituent of SDC, in the current research, we limited our study to simply reaching the predefined maximum number of iterations as the convergence condition and postponed the other possibilities to future work.

To summarize, so far, we proposed two update rules, the vanilla and ADAM update rules, and this led us to propose two SDC clustering methods, for which we represent them as VSDC and ASDC, respectively. We summarize the SDC methods in common steps in Algorithm (1).

Algorithm 1: Steepest Descent Clustering (SDC)

Input: $D = \{X, A\}$: Feature-rich network; K : clusters' number; the choice of update rule.

Hyperparameters: α : step size; T : maximum number of iterations; β_1 and β_2 : first and second moments decay coefficients.

Result: $S = \{s_k^{(t)}\}_{k=1}^K$ % set of K binary cluster membership vectors;

$C = \{c_k^{(t)}\}_{k=1}^K$ % set of K centroids in feature space;

$\Lambda = \{\lambda_k^{(t)}\}_{k=1}^K$ % set of K centroids in network space.

Initialize: Randomly initialize C, Λ and S .

for $t \in \text{Range}(T)$ **do**

for $(x_i, a_i) \in D$ **do**

 find k using Eqn. (2) and set i -th entry of the $s_k^{(t)}$ to one;

 update the centroids using the corresponding equations, i.e., (3) or (4);

end

end

Our reported experiments in section (5) show that even the tuned SDC algorithms did not obtain promising results. We empirically investigated the possibilities of obtaining such results, and we noticed two issues with SGD methods: (i) the quality of the seeds to initialize the centroids and (ii) the quality and/or the order of the data points in the batches. While the former is a well-known limit of the gradient descent optimization approach, the latter, to the best of our knowledge, did not receive any attention from research communities. We tried various seed initialization methods to tackle (i), and unfortunately, none provided us with decent solutions; thus, we postponed it to our future study and merely adopted the advice of using various seeds and selecting the best-obtained result. However, we managed to tackle (ii) by refining the batches as follows.

We assumed that ideally, the so-called appropriate order of data points should consist of data points for which the magnitude of the gradients changes as smoothly as possible during the update process; furthermore, those data points for which their gradients' magnitude is significantly larger than others should not be included during the centroids updates. One may consider such data points, with significantly larger gradients' magnitudes, as anomalies and exclude them; however, in our opinion, this is a subject to another research and we did not exclude such data points from the cluster recovery results. To refine the batches,

we relied on Bootstrapping to derive the empirical distributions of the gradients' magnitudes, i.e., the average, μ , and the standard deviation, σ . After that, we introduce a refinement coefficient, τ , and we update the corresponding centroids if the magnitude of the gradients lies between $\mu \pm \tau\sigma$. Obviously, the larger the τ the softer the centroids update refinement condition. We outline our proposed refined steepest descent clustering (RSDC) in the algorithm (2).

Algorithm 2: Refined Steepest Descent Clustering (RSDC)

Input: $D = \{X, A\}$: Feature-rich network; K : clusters' number.
Hyperparameters: α : step size; T : maximum number of iterations; τ : the refinement coefficient.
Result: $S = \{s_k^{(t)}\}_{k=1}^K$ % set of K binary cluster membership vectors;
 $C = \{\mathbf{c}_k^{(t)}\}_{k=1}^K$ % set of K centroids in feature space;
 $A = \{\boldsymbol{\lambda}_k^{(t)}\}_{k=1}^K$ % set of K centroids in network space.
Initialize: Randomly initialize C, A and S .
 $G = []$; % an empty list to track the history of gradients' magnitude
for $t \in \text{Range}(100)$ **do**
 $B = \{b_i\}_{i=1}^N$; % random indices with replacement from $\{1, \dots, N\}$
 $g = []$; % an empty list to track the history of gradients' magnitude
 for $b_i \in B$ **do**
 find k using Eqn. (2) and set i -th entry of the $s_k^{(t)}$ to one;
 update the centroids using the equation (3) and its equivalent in the network space;
 append $\|\nabla_{c_k} f(\cdot)\| + \|\nabla_{\lambda_k} f(\cdot)\|$ to g
 end
 compute the mean of g and append it to G
end
compute average μ , and standard deviation σ of G ;
for $t \in \text{Range}(T)$ **do**
 for $(\mathbf{x}_i, \mathbf{a}_i) \in D$ **do**
 find k using Eqn. (2) and set i -th entry of the $s_k^{(t)}$ to one;
 if $\mu - \tau * \sigma \leq \|\nabla_{c_k} f(\mathbf{c}_k, \mathbf{x}_i)\| + \|\nabla_{\lambda_k} h(\boldsymbol{\lambda}_k, \mathbf{a}_i)\| \leq \mu + \tau * \sigma$ **then**
 update the centroids using the Eqn. (3) and its equivalent in the network space;
 end
 end
end

The introduced refinement condition not only has a desirable refinement impact on the centroids updates but also has a smoothing impact on the centroids updates, that is, after several updates or even a couple of iterations, those data points that did not satisfy the refinement condition previously, may satisfy it later due the previous centroid updates by the smoother data points. Although the adaptive adjustment of τ seems to be a plausible solution, in the current research, we empirically study the impact of fixed τ on various synthetic data

Table 1. Real-world data sets: symbols N, E, and F stand for the number of nodes, the number of edges, and the number of node features, respectively.

Name	Nodes	Edges	Features	Number of Communities	Ground Truth	Ref
Malaria HVR6	307	6526	6	2	Cys Labels	[14]
Lawyers	71	339	18	6	Derived out of office and status features	[15]
Parliament	451	11646	108	7	Political parties	[2]
COSN	46	552	16	2	Region	[7]

and propose a default value for it. Last but not least, in the current study, since the vanilla update rule has fewer number of hyperparameters, we limited the refined steepest descent update rule to the vanilla version; nevertheless, applying this process to the other update rules, like ADAM, could be an intriguing future work and we postponed it to another paper.

Implementing our proposed methods using JAX, an automatic differentiation library [3], enables the users to apply any differentiable distance function of their choice to improve the cluster recovery results. The source code of SDG methods is publicly available at <https://github.com/Sorooshi/RSDC>.

3 Experimental Setting

3.1 Algorithms Under Comparison

We selected our competitors, among the previous papers published in well-established journals (indexed in the top two quarterlies of the Scopus or the Web of Science), concerning the two following criteria: (a) the well-documented implementation of the the proposed method should be publicly available, and (b) the proposed method has a reasonable number of hyperparameters. With these criteria, we compared the performance of our proposed methods with three state-of-art algorithms, namely, DMoN [17], a method based on graph convolutional neural networks, EVA [6], a heuristic method using modularity and purity, and KEFRIN [22], an extension of K-Means method, from the literature. We have extensively assessed all methods under consideration in computational experiments.

3.2 Data Sets

We tested the performance of the methods under consideration using both real-world and synthetic data sets. We describe them in the following subsections.

Real World Data Sets. We use four real-world data sets and summarize them in Table 1. For more details, readers may refer to [20].

Generating Synthetic Data Sets. We utilized the synthetic data generation mechanism proposed in [21], which later were used in [16, 20, 22]. The readers can refer to any of them for more details. Here, we briefly describe its different components for generating the networks and the categorical features.

Network Data Generation. Given the number of nodes N and the number of clusters K , the clusters' cardinalities are defined randomly so that no community has less than a pre-defined number of nodes (in our experiments, we set this number to 30), and the sum of the cardinalities sums to N . We consider three settings for N : (a) $N = 200$ for small-size networks, (b) $N = 1000$ for medium-sized networks, and (c) $N = 400$ for hyperparameter-tuning networks. We postponed the analysis of large networks to another research.

We populate the clusters with nodes' indices defined by the cluster cardinalities. Then, we specify two probability values, p and q . Every within-community link is drawn with the probability p , independently of other links. Similarly, every between-community link is drawn independently with the probability q .

Categorical Features Generation. To model categorical attributes, first the number of subcategories for each category is randomly chosen from the set $\{2, 3, \dots, L\}$ where $L = 10$ for small-size networks and $L = 15$ for the medium-size networks. Next, given the number of clusters, K , and the numbers of entities in each cluster, N_k for $(k = 1, \dots, K)$; the cluster centers are generated randomly so that no two centers may coincide at more than 50% of features.

Once a center of k -th cluster, $c_k = (c_{kv})$, is specified, N_k entities of this cluster are generated as follows. Given a pre-specified threshold of intermix, ϵ between 0 and 1, for every pair (i, v) , $i = 1 : N_k$; $v = 1 : V$, a uniformly random real number r between 0 and 1 is generated. If $r > \epsilon$, the entry x_{iv} is set to be equal to c_{kv} ; otherwise, x_{iv} is taken randomly from the set of subcategories specified for feature v . Consequently, all entities of the k -th cluster coincide with its center, up to rare errors if ϵ is large enough. The larger the epsilon, the more diverse, and thus intermixed, would be the generated entities.

3.3 Evaluation Criteria

We used the Adjusted Rand Index [10] (ARI) as our evaluation criterion. The Eq. 7 explains this criterion:

$$ARI(S, T) = \frac{\sum_{k,l} \binom{n_{kl}}{2} - [\sum_k \binom{a_k}{2} \sum_l \binom{b_l}{2}]/\binom{N}{2}}{\frac{1}{2}[\sum_k \binom{a_k}{2} + \sum_l \binom{b_l}{2}] - [\sum_k \binom{a_k}{2} \sum_l \binom{b_l}{2}]/\binom{N}{2}} \quad (7)$$

where $a_k = \sum_{l=1}^L n_{kl} = |S_k|$ and $b_l = \sum_{k=1}^K n_{kl} = |T_l|$, are the marginal row and marginal column of the contingency table, for the given clustering results, $S = \{S_1, S_2, \dots, S_K\}$, and the ground truth $T = \{T_1, T_2, \dots, T_L\}$. The closer the value of ARI to one, the better is the match between the clustering results and the ground truth; ARI=1.0 shows that $S = T$, i.e., the perfect clustering recovery results. If one of the partitions consists of just one part, then ARI=0. Cases at which ARI is negative may occur and its lower bound is not known.

4 Scrutinizing the Main Hyperparameters of the Proposed Methods

We scrutinized the impact of (i) the step size, α , (ii) the first and second moments decay coefficients, β_1 and β_2 , in the feature space and the network space, and (iii) the refinement coefficient τ on 60 synthetic feature-rich networks with 400 nodes, six quantitative features (refer to [16] for more details), and four communities and various cluster intermixes, within- and between-communities probabilities. However, due to the limited length, we could not present those computational results and only summarized their tuned values in Table 2.

Table 2. Best tuned hyperparameter values.

update rule		step size (α)	β_1	β_2	τ
VSDC		0.0001	–	–	
ASDC	Feature Space	[0.0001 – 0.01]	[0.3, 0.6]	[0.9, 0.99]	–
	Network Space		[0.6, 0.9]		
RSDC	Feature Space	0.1	–	–	[0.3, 0.9]
	Network Space				

We fixed the step size in VSDC, ASDC, and RSDC to 0.0001, 0.001, and 0.1 respectively. We consider the median of demonstrated ranges for the first and second moments decaying coefficients as the optimal values in ASDC. Finally, for the refinement coefficient, τ , in the RSDC method, we made a conservative choice and fixed it to 0.3 in the rest of our computations.

5 Experimental Results

5.1 Comparison over Real-World Data Sets

We compared the performance of the proposed methods over four real-world data sets with three state-of-the-art algorithms and reported the results in Table 3. Due to our flexible implementation using JAX we used both Euclidean and cosine distance metrics in SDC methods and reported the best-obtained results. More precisely, except for RSDC at HVR and COSN data sets, for which we used cosine distance, the rest of the reported SDC methods’ results were obtained using Euclidean distance.

RSDC is the overall winner of real-world competitions by winning three out of four cases. DMoN won the lawyers competition, and KEFRiN jointly with RSDC won the COSN competition.

5.2 Comparison over Synthetic Data with Categorical Features

We reported the results of comparison over small-size and medium-size synthetic feature-rich networks in Table 4 and 5, respectively.

Although RSDC is the winner of synthetic feature-rich networks, and its performance is remarkably better performances than its counterparts; however, surprisingly, it loses its efficiency in the last two rows of these two tables. Studying the reason behind such poor performance is an ongoing research.

Table 3. Real-world data sets comparison: average and standard deviation of ARI over ten random initialization.

data set	DMoN	EVA	KEFRiN	VSDC	ASDC	RSDC
HRV6	0.64 ± 0.00	0.036 ± 0.004	0.69 ± 0.38	0.507 ± 0.182	0.246 ± 0.403	0.848 ± 0.012
Lawyers	0.60 ± 0.04	0.159 ± 0.028	0.44 ± 0.14	0.360 ± 0.107	0.323 ± 0.060	0.414 ± 0.053
Parliament	0.48 ± 0.02	0.005 ± 0.001	0.41 ± 0.05	0.398 ± 0.070	0.492 ± 0.076	0.561 ± 0.038
COSN	0.91 ± 0.00	-0.004 ± 0.000	1.000 ± 0.000	0.731 ± 0.094	0.694 ± 0.108	1.000 ± 0.000

Table 4. Small-size synthetic data with categorical features: average and standard deviation of ARI over ten random initialization.

p, q, ϵ	EVA	DMoN	KEFRiN	VSDC	ASDC	RSDC
0.9, 0.3, 0.9	0.185 ± 0.046	0.709 ± 0.101	0.922 ± 0.119	0.836 ± 0.160	0.802 ± 0.094	0.978 ± 0.057
0.9, 0.3, 0.7	0.211 ± 0.053	0.380 ± 0.107	0.819 ± 0.142	0.885 ± 0.118	0.844 ± 0.125	0.975 ± 0.053
0.9, 0.6, 0.9	0.266 ± 0.080	0.412 ± 0.109	0.726 ± 0.097	0.232 ± 0.044	0.018 ± 0.042	0.958 ± 0.014
0.9, 0.6, 0.7	0.321 ± 0.060	0.213 ± 0.051	0.711 ± 0.145	0.192 ± 0.048	0.003 ± 0.002	0.916 ± 0.038
0.7, 0.3, 0.9	0.126 ± 0.039	0.566 ± 0.105	0.877 ± 0.130	0.461 ± 0.100	0.299 ± 0.160	0.998 ± 0.004
0.7, 0.3, 0.7	0.126 ± 0.025	0.292 ± 0.077	0.795 ± 0.117	0.464 ± 0.096	0.286 ± 0.107	0.992 ± 0.012
0.7, 0.6, 0.9	0.015 ± 0.015	0.345 ± 0.064	0.834 ± 0.132	0.031 ± 0.010	0.002 ± 0.002	0.037 ± 0.011
0.7, 0.6, 0.7	0.008 ± 0.007	0.115 ± 0.058	0.540 ± 0.107	0.024 ± 0.01	0.003 ± 0.002	0.026 ± 0.007

Table 5. Medium-size synthetic data with categorical features: average and standard deviation of ARI over ten random initialization.

p, q, ϵ	EVA	DMoN	KEFRiN	VSDC	ASDC	RSDC
0.9, 0.3, 0.9	0.121 ± 0.031	0.512 ± 0.137	0.724 ± 0.097	0.388 ± 0.210	0.001 ± 0.000	0.922 ± 0.023
0.9, 0.3, 0.7	0.076 ± 0.038	0.272 ± 0.073	0.742 ± 0.182	0.475 ± 0.112	0.018 ± 0.052	0.925 ± 0.038
0.9, 0.6, 0.9	0.159 ± 0.046	0.370 ± 0.063	0.652 ± 0.110	0.001 ± 0.001	0.001 ± 0.001	0.718 ± 0.050
0.9, 0.6, 0.7	0.109 ± 0.046	0.168 ± 0.030	0.733 ± 0.083	0.002 ± 0.001	0.001 ± 0.000	0.647 ± 0.062
0.7, 0.3, 0.9	0.078 ± 0.036	0.446 ± 0.099	0.641 ± 0.111	0.007 ± 0.014	0.001 ± 0.000	0.916 ± 0.015
0.7, 0.3, 0.7	0.059 ± 0.010	0.228 ± 0.077	0.797 ± 0.088	0.002 ± 0.004	0.001 ± 0.000	0.932 ± 0.016
0.7, 0.6, 0.9	0.002 ± 0.002	0.332 ± 0.051	0.591 ± 0.094	0.001 ± 0.001	0.001 ± 0.002	0.011 ± 0.006
0.7, 0.6, 0.7	0.002 ± 0.002	0.133 ± 0.016	0.773 ± 0.070	0.001 ± 0.000	0.001 ± 0.000	0.015 ± 0.015

6 Conclusion and Future Work

In this work, inspired by the triumph of gradient descent and automatic differentiation approach in supervised learning tasks of machine learning, we proposed a novel clustering methodology to detect the clusters in feature-rich networks. Since the gradient is the steepest descent direction in the optimization process and is the core of our proposed method, we named the base of our proposed method steepest descent clustering (SDC). The gradient descent update rule in its vanilla form is prone to slow convergence close to valley floors: and to address this issue, we utilized the adaptive moment estimation update rule. We distinguished between the two versions of SDC concerning the update rules, i.e., the vanilla update rule (VSDC) and the ADAM-based update rule (ASDC). Our experiments using VSDC and ASDC did not lead to promising results.

We noticed two issues with their performances: their sensitivity to (i) the seed initialization and (ii) the quality and/or the order of the data points in the batches. We could not tackle the former issue, and merely adopted the standard practice of initialization with different seeds and selecting the best-obtained results. Unlike the former problem, we tackled the second problem by relying on Bootstrapping to derive the empirical distribution of gradients' magnitude and introducing a refinement condition to update the centroids accordingly. We named this version of our proposed method refined steepest descent clustering (RSDC).

We empirically validated and compared the performance of SDC methods over four real-world data sets and 160 synthetic feature-rich networks with categorical features. Our experiments proved that the refined steepest descent method is effective and in the majority of the competitions appeared to be the winner.

The current research is not without limits, and those limits form our future work directions. First, the SDC methods are sensitive to seed initialization: proposing a technique to reduce this sensitivity or initialization of the seeds more effectively is one of the most important future directions. Extending our experiments to a broader range of synthetic and real-world feature-rich networks, is our second future direction. Conducting more experiments using different distance metrics, like Canberra distance, to improve the cluster recovery results, can be considered as another future direction. Studying the impact of the number of iterations on the performance of SDC methods to propose a generic rule can be considered as another future direction.

References

1. Al-Andoli, M.N., Tan, S.C., Cheah, W.P., Tan, S.Y.: A review on community detection in large complex networks from conventional to deep learning methods: a call for the use of parallel meta-heuristic algorithms. *IEEE Access* **9**, 96501–96527 (2021)
2. Bojchevski, A., Günnemann, S.: Bayesian robust attributed graph clustering: Joint learning of partial anomalies and group structure. In: *Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 1–10. AAAI Press, California, USA (2018)

3. Bradbury, J., et al.: JAX: composable transformations of Python+NumPy programs (2018). <http://github.com/google/jax>
4. Choong, J.J., Liu, X., Murata, T.: Optimizing variational graph autoencoder for community detection with dual optimization. *Entropy* **22**(2), 197 (2020)
5. Chowdhury, S., Needham, T.: Generalized spectral clustering via gromov-wasserstein learning. In: *International Conference on Artificial Intelligence and Statistics*, pp. 712–720. PMLR (2021)
6. Citraro, S., Rossetti, G.: Identifying and exploiting homogeneous communities in labeled networks. *Appl. Netw. Sci.* **5**(1), 1–20 (2020)
7. Cross, R., Parker, A.: *The Hidden Power of Social Networks: Understanding How Work Really Gets Done in Organizations*, 1st edn. Harvard Business Press, USA (2004)
8. Fang, W., Wang, X., Liu, L., Wu, Z., Tang, S., Zheng, Z.: Community detection through vector-label propagation algorithms. *Chaos, Solitons Fractals* **158**, 112066 (2022)
9. Hu, F., Liu, J., Li, L., Liang, J.: Community detection in complex networks using node2vec with spectral clustering. *Phys. A Stat. Mech. Appl.* **545**, 123633 (2020)
10. Hubert, L., Arabie, P.: Comparing partitions. *J. Classif.* **2**, 193–218 (1985)
11. Kim, W., Kanazaki, A., Tanaka, M.: Unsupervised learning of image segmentation based on differentiable feature clustering. *IEEE Trans. Image Process.* **29**, 8055–8068 (2020)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014). arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
13. Kochenderfer, M.J., Wheeler, T.A.: *Algorithms for Optimization*. MIT Press (2019)
14. Larremore, D.B., Clauset, A., Buckee, C.O.: Network approach to analyzing highly recombinant malaria parasite genes. *PLoS Comput. Biol.* **9**(10), e1003268 (2013)
15. Lazega, E.: *The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership*, 1st edn. Oxford University Press, GB (2001)
16. Mirkin, B., Shalileh, S.: Community detection in feature-rich networks using data recovery approach. *J. Classif.* **39**(3), 432–462 (2022)
17. Müller, E.: Graph clustering with graph neural networks. *J. Mach. Learn. Res.* **24**, 1–21 (2023)
18. Nesterov, Y.E.: A method of solving a convex programming problem with convergence rate $O(k^{-2})$. In: *Doklady Akademii Nauk*, vol. 269, pp. 543–547. Russian Academy of Sciences (1983)
19. Shalileh, S.: An effective partitional crisp clustering method using gradient descent approach. *Mathematics* **11**(12), 2617 (2023)
20. Shalileh, S., Mirkin, B.: Least-squares community extraction in feature-rich networks using similarity data. *Plos One* **16**(7), e0254377 (2021)
21. Shalileh, S., Mirkin, B.: Summable and nonsummable data-driven models for community detection in feature-rich networks. *Soc. Netw. Anal. Min.* **11**, 1–23 (2021)
22. Shalileh, S., Mirkin, B.: Community partitioning over feature-rich networks using an extended k-means method. *Entropy* **24**(5), 626 (2022)
23. Wilson, D.R., Martinez, T.R.: The general inefficiency of batch training for gradient descent learning. *Neural Netw.* **16**(10), 1429–1451 (2003)
24. Ye, F., Chen, C., Zheng, Z.: Deep autoencoder-like nonnegative matrix factorization for community detection. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 1393–1402 (2018)
25. Zeiler, M.D.: ADADELTA: an adaptive learning rate method (2012). arXiv preprint [arXiv:1212.5701](https://arxiv.org/abs/1212.5701)



Detecting Strong Cliques in Co-authorship Networks

Lukas Papik^(✉), Eliska Ochodkova, and Milos Kudelka

Department of Computer Science, VSB - Technical University of Ostrava, Ostrava,
Czech Republic

lukas.papik.st@vsb.cz

Abstract. The study of complete sub-graphs belongs to the classical problems of graph theory. Thanks to sociology, the term clique has come to be used for structures representing a small group of people or other entities who share common characteristics and know each other. Clique detection algorithms can be applied in all domains where networks are used to describe relationships among entities. That is not only in social, information, or communication networks but also in biology, chemistry, medicine, etc. In large-scale, e.g., social networks, cliques can have hundreds or more nodes. On the other hand, e.g., in co-authorship networks representing publishing activities of groups of authors, cliques contain, at most, low dozens of nodes. Our paper describes experiments on detecting strong cliques in two weighted co-authorship networks. These experiments are motivated by the assumption that not every clique detected by traditional algorithms truly satisfies the sociological assumption above. Informally speaking, the approach presented in this paper assumes that each pair of clique nodes must be closer to each other and other clique nodes than to non-clique nodes. Using experiments with weighted co-authorship networks, we show how clique detection results differ from the traditional approach when both the strength of the edge (weight) and the structural neighborhood of the clique are considered simultaneously in the analysis.

Keywords: clique detection · weighted network · co-authorship network · strong clique

1 Introduction

One of the tasks of network analysis is to identify cohesive groups of nodes, which are subsets of nodes with relatively strong or frequent ties [19] that are useful for exploring the local structure of networks. The primary structures examined are cliques, plexes, cores, etc.

In the field of clique detection in networks, we find mainly tasks related to so-called maximal cliques that are not contained in another clique and thus cannot be extended by any other node. The first task is to find the *maximum clique*,

i.e., the clique with the largest number of nodes in the network; a modification of this task is to find the maximum weighted clique with the largest sum of edge weights inside the clique [9]. The second task is *listing all maximal cliques* [2]. The cliques found may vary in size (from dyads and triads to groups with many nodes), and some of them may overlap.

Cliques are structures that can be significantly separated from the rest of the network in some cases. In other cases, they may be the core of larger communities that may be unique. In such cases, the communities have a so-called core-periphery structure [4], in which the clique forms the so-called core, and the other nodes are part of the periphery of the community. In biology, we can find cliques, e.g., in protein complexes formed by interacting proteins in so-called PPI (protein-protein interactions) networks [14].

The definition of a clique is very strict. However, thanks to communication technologies, the linking between nodes is simplified, and nodes that can be seen more as randomness or noise can be part of a clique. For example, a clique of three authors who have collectively written more than a dozen publications will be joined over time by four other authors, but each of them will have only one publication. If we search for all maximal cliques in this case, a clique with seven nodes will be detected. From the point of view of the analysis, it is more important to find the core of this structure, i.e., the clique formed by the three strongly connected authors. The neighborhood of this clique can be studied in the next step of the analysis.

The requirement of the occurrence of every possible edge in a clique is stringent, so we can naturally relax this requirement and consider other dense structures like k -plexes (a maximum subset of n nodes in the network such that each node is connected to at least $n - k$ others), k -cores (a maximum subset of nodes such that each one is connected to at least k other nodes in this subset) or k -cliques (a maximum subset of nodes such that each is no more than a distance k away from any of the others via the edges of the network). If we restrict ourselves to paths between nodes that run only within the subset, we call the resulting objects either k -clans or k -clubs [15].

Our experiments focus on co-authorship networks whose structure reveals many interesting features of academic communities. Researchers can study many interesting problems [11, 17] like discovering the most important authors, what the largest communities of co-authors look like, the cohesiveness of communities, and how publication patterns vary across research areas and over time. These and similar questions can be answered by analyzing the basic structural properties of networks [16] (number of papers per author, number of authors per paper, distances between authors, clustering coefficient, assortativity coefficient, community structure, etc. [12]). Other areas of interest in analyzing co-authorship networks are predicting collaborations [3, 6] or examining changes in collaboration patterns over time [18].

In the approach presented below, we aim to detect strong cliques in co-authorship networks that do not contain noise due to the randomness of the linkage. The detected cliques can be seen as cores of clusters or communities.

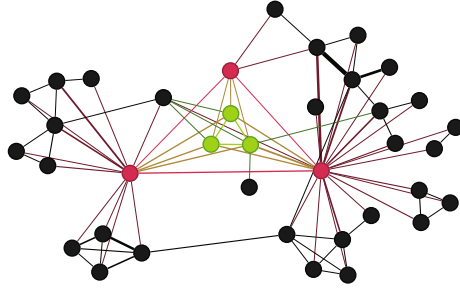


Fig. 1. Maximal vs. strong clique. The six colored nodes form a clique, but the three red nodes have a stronger connection outside the strong clique formed by the three green nodes. In the context shown, the red nodes can be seen as hubs providing connections from the nodes of a strong clique to other parts of the network.

2 Strong Cliques

The question is what options we have for detecting a strong clique, the formal definition we will give later. Assuming that the network is weighted, we can choose a threshold for weights and keep only edges above this threshold for further analysis. The main problem with this approach is that the threshold should vary in different parts of the network; it should be lower for nascent cliques than for cliques based on long-term cooperation or interaction. Moreover, this approach cannot be applied to unweighted networks or where a simple and unambiguous interpretation of weights cannot be found. Therefore, it is essential to examine both the connection between clique nodes and their neighborhood, as well as the interconnection strength between nodes when detecting a strong clique. Thus, on the one hand, we want to label as strong cliques those with weaker interconnections but no (or very weak) connections to their neighborhood. On the other hand, we do not want to consider as strong those cliques that have strong interconnection but even stronger connections to their neighborhood (Fig. 1).

2.1 Structural Dependency

To calculate the value of the connectivity between pairs of nodes combining the weight and the analysis of the structure of their neighborhood, we use the so-called structural dependency [10]. Unless otherwise stated, we will use the word dependency for this structural dependency hereafter. In determining the dependency, we assume that the strength of the relationship between two adjacent nodes A, B is determined not only by the weight of their edge $w(A, B)$. On the one hand, we must take into account how many common neighbors and how strongly they are connected to, and on the other hand, how many and how strongly each of them is connected separately. We then define the dependency $D(A, B)$ of node A on node B as follows:

$$D(A, B) = \frac{w(A, B) + \sum_{X_i \in \Gamma(A, B)} w(A, B, X_i)}{\sum_{X_i \in N(A)} w(A, X_i)} \quad (1)$$

The weight of the relationship between two nodes A, B given by their common neighbor X is

$$w(A, B, X) = w(B, A, X) = \frac{w(A, X) \cdot w(B, X)}{w(A, X) + w(B, X)} \quad (2)$$

$\Gamma(A, B)$ is the set of common neighbors of nodes A, B , and $N(A)$ is the neighborhood (set of all neighbors) of node A . A dependency defined in this way is non-symmetric; although the values of the numerators are the same in both directions of the dependency, the values of the denominators may be different. Therefore, the interdependencies between nodes A, B may be substantially different. It is important to note here that the non-symmetry holds even in unweighted networks, and in the computation, for each common neighbor X_i , the $w(A, B, X_i)$ is 0.5. Thus, the value of the dependency between nodes A, B is most affected by the number of common triangles compared to their other edges to the neighborhood of the clique. Between two nodes three situations can occur (see Fig. 2): (2a) they are independent of each other and look like members of two different communities, (2b) they are one-way dependent and the left node has a stronger dependency to other nodes and (2c) although the nodes have no common neighbors, they are mutually dependent because weight between them is high.

Definition (Strong Clique). Let us have a threshold $\tau \in (0, 1]$. A strong clique is a clique in which each pair of nodes A, B is mutually dependent, and the following holds:

$$D(A, B) \geq \tau \wedge D(B, A) \geq \tau \quad (3)$$

The lower the threshold τ , the more benevolent the assessment of mutual dependency. Conversely, for $\tau = 1$, mutual dependency occurs only for an isolated pair of nodes. For example, the value $\tau = 0.5$ is quite strict because edges in the neighborhood of nodes have more influence than edges with common neighbors in the dependency calculation. Note that each strong clique is either maximal or contained in one or more overlapping maximal cliques.

2.2 Dependency Threshold Estimation

The threshold τ raises the question of how to set this threshold, as each authorship network may differ. Suppose we take an edge from a network connecting nodes a and b ; for this edge, we can compute two dependencies, one as $D(a, b)$ and the other as $D(b, a)$, because of the asymmetry of the dependency. It is worth noting that these dependencies are from the original network, as, at this moment, we do not know the threshold for removing edges.

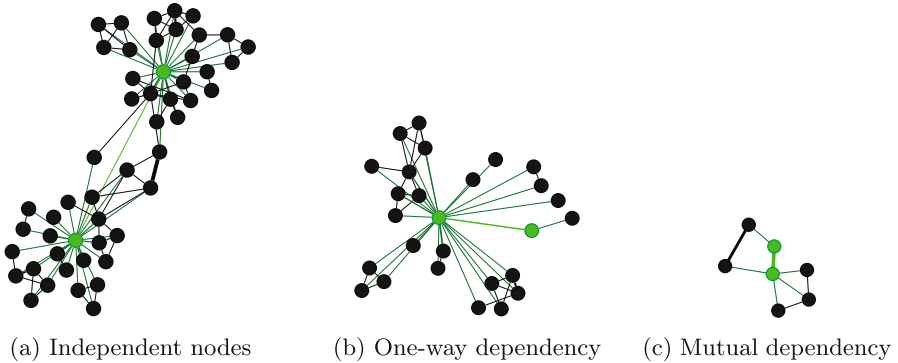


Fig. 2. Dependency situations between two nodes that may occur. Nodes are independent of each other (2a). One-way dependency, where the node on the right is dependent on the node on the left, but not vice versa (2b). Mutual dependency is where nodes are dependent on each other (2c).

Let us refer to the cliques found in the networks as reference cliques. Thus, individual reference cliques may have authors with low dependency on other authors within the clique. Let us take all edges from each reference clique in the network and find the minimum and maximum edge dependency of each clique. We can average these minimum and maximum values across all reference cliques.

The minimum dependency within a clique means that if we set the dependency threshold to this value or less, we would get the same clique. This is because each edge within the clique satisfies the strong clique condition (3) so that no edge would be removed. Similarly, for maximum dependency within a clique, if we set the dependency threshold above this value, the reference clique would disappear because all edges would be removed, as no edge from the clique satisfies the strong clique condition (3). Thus, the dependency threshold should be on average in this interval, since we know the references and their structure. The estimated dependency threshold can be set halfway between the average minimum and maximum as a balanced value, but it should be taken as a sort of starting point that we can then adjust as needed.

3 Experiments

For the experiments, we used two undirected weighted co-authorship networks in which the edge weights represent the number of joint publications. The co-authorship networks are sparse and low-rank and thus contain relatively strongly formed cliques involving jointly publishing members of more or less formal teams. The first network *GEOM* representing collaboration in computational geometry¹ has 6158 nodes and 11898 edges. For example, this network has been used for link prediction [13] or for clustering [7, 8]. The network has also been used for

¹ <https://networkrepository.com/geom.php>.

core detection [1], but this approach differs from ours; we do not evaluate the importance of individual nodes, but the relationship between two nodes represented by the dependency. The second one *CHI* is a co-authorship network for the International Conference on Human Factors in Computing Systems² with 20046 nodes and 54111 edges. Authors used the network in experiments for visualizing collaboration relations as geographic-like maps [5].

Both co-authorship networks were pre-processed to result in six new networks for each. In the first network, *Net1*, we took only the largest connected component and removed those nodes with a degree of 1. In the second network, *Net2*, we first removed edges with a weight of 1 (potential noise) and then took the largest connected component without nodes with a degree of 1. In the third and fourth networks, *Net1-D3* and *Net1-D5*, we preserved only those edges from the *Net1* network whose nodes had a mutual dependency of at least $\tau = 0.3$ and $\tau = 0.5$ (the dependency was computed in the original networks). We did the same for the *Net2* network and obtained the *Net2-D3* and *Net2-D5* networks. The threshold for dependency was chosen to be $\tau = 0.3$ because both *Net1* and *Net2* networks had an average between their average minimum and maximum dependencies of around 0.3. Setting $\tau = 0.5$ was chosen to demonstrate the impact of a higher threshold on the networks. GEOM *Net1* and *Net2* have an average minimum dependency 0.07 and an average maximum dependency of 0.6 and 0.55. *CHI Net1* and *Net2* have an average minimum dependency 0.08 and 0.11 and an average maximum dependency of 0.62 and 0.51.

After pre-processing the networks, the Bron-Kerbosch algorithm [2] was applied to list all maximal cliques in each network. Next, we analyzed the clique distribution and the impact of the dependency threshold on each network.

3.1 Results and Discussion

For each of the twelve networks, the distribution of detected cliques by size in one of the four plots is shown (see Fig. 3). Each plot contains a comparison of cliques detected (1) without accounting for mutual dependency, (2) with mutual dependency with a threshold $\tau = 0.3$, and (3) with mutual dependency with a threshold $\tau = 0.5$. The expected result is that significantly more cliques are detected in the networks when we do not work with mutual dependency. However, it can be noticed that if we take mutual dependency into account, much fewer dyads are detected at the expense of triads. This would suggest that the triads are often, in fact, dyads complemented by a hub (as explained by the comment in Fig. 1), but we will show later that this does not apply only to triads. The use of mutual dependency allows us to detect these situations (detection of hubs is explained in [10]).

Other observations provide a natural interpretation. The stricter we are on the mutual dependency, the smaller the detected cliques and with a smaller size. This is mainly because the larger the clique, the higher the chance that one or more hubs are part of the clique.

² <http://gmap.cs.arizona.edu/datasets>.

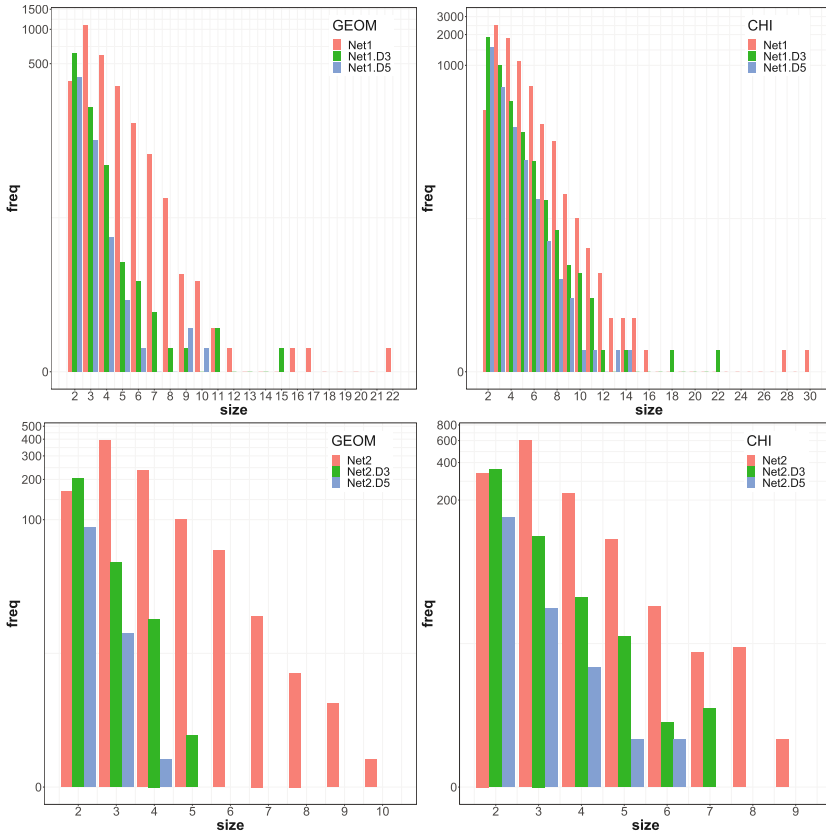


Fig. 3. Distribution of cliques in the studied networks. The left/right column shows the distributions of cliques according to their size detected in the pre-processed components of the GEOM/CHI network.

3.2 Dependency Threshold Effect

Fig. 4 shows that for both Net1 and Net2 from both networks the number of cliques that have edges with low minimal dependencies (below 0.1) are the most frequent for each size. This means that cliques have one or more pair of nodes that don't share much together. One of the node might be a hub, as we mentioned before, or have a stronger connection to some other clique. Edges like these can be removed using the optimal dependency threshold.

Two scenarios can occur when an edge from a clique is removed: first, the clique disappears; second, and more often, the clique splits into smaller sub-cliques. Obviously, the more nodes the clique has, the smaller and more tightly connected sub-groups it must have within it; smaller groups of people may write together more often because they are more related to each other than to the group as a whole. This behavior can be seen in Figs. 5 and 6, where we created an

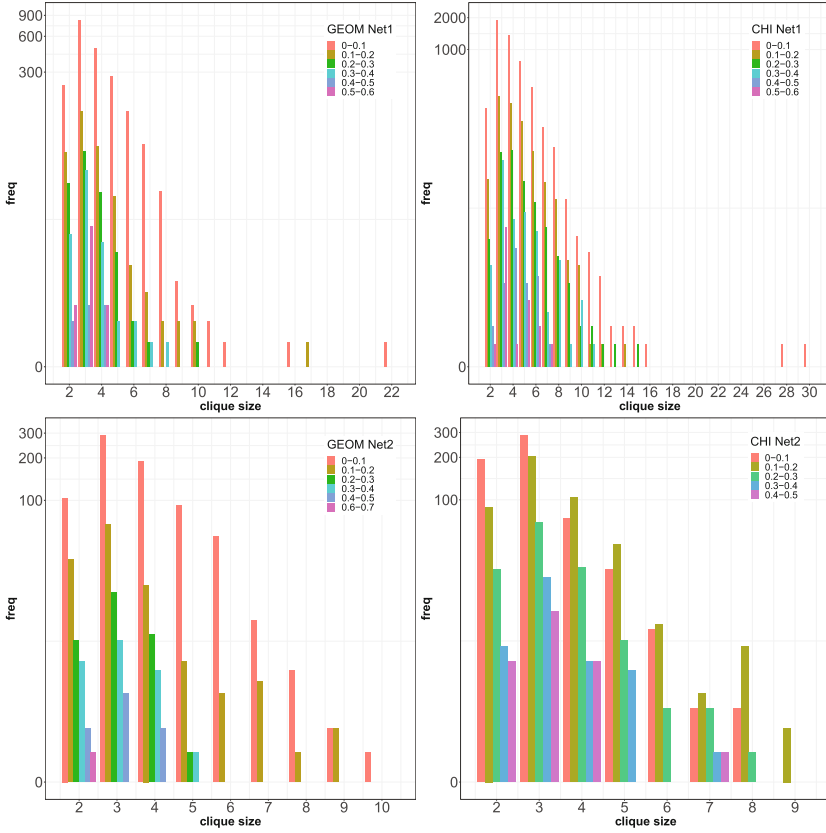


Fig. 4. Distribution of cliques minimum dependency according to their size

induced subgraph from the clique nodes from the network, where the dependency threshold is applied. Next, we enumerate all maximal cliques in this subgraph and sum the sizes of these cliques. Most of the reference cliques for networks Net1 and Net2 and after applying both dependencies are reduced to dyads only; this explains why there are so many dyads in Fig. 3. However, the larger the clique is, the larger the sub-cliques can be found.

On the other hand, the interesting thing to notice are cliques of size 2. Compared to cliques of different sizes, it can be seen in Fig. 7 that for Net1 and Net2 from both networks, the cliques of size 2 have a lower average maximum dependency; in other words, the dependency between dyads is low. There may be many real-world explanations for why this happens. From a topological point of view, we can look at the possible situations in Fig. 2a, 2b and 2c. If the threshold of dependency were set to 0.3, we would obtain only a small number of original dyads. It indicates that the last figure is uncommon in our networks due to the low average maximum dependency. That means that two authors mostly do not

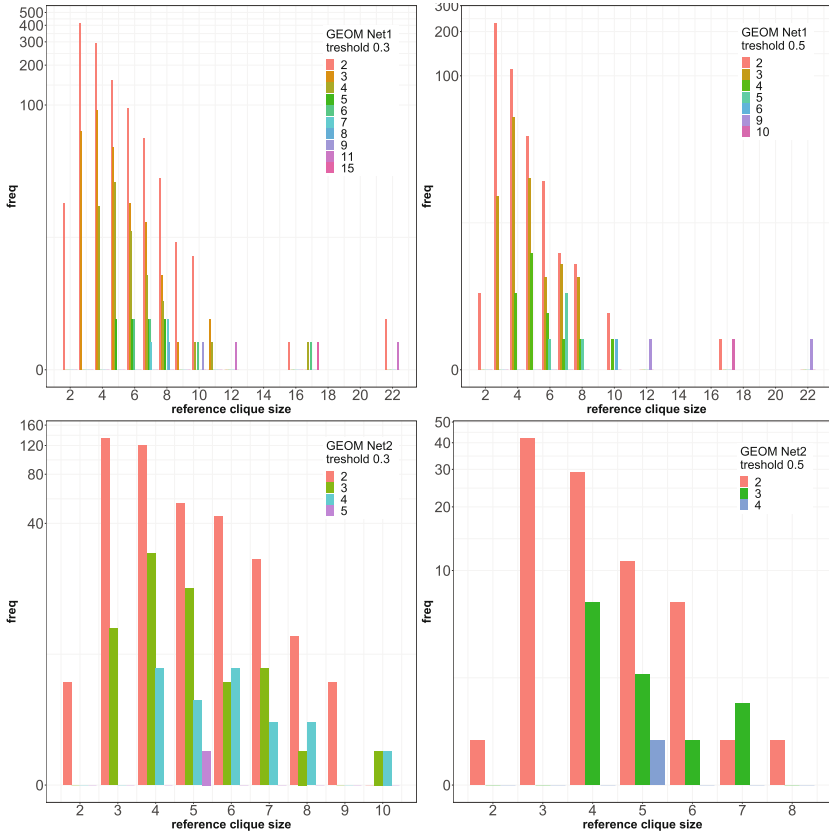


Fig. 5. Sub-clique size distribution after applying the dependency thresholds for GEOM Net1 and Net2.

write together on their own, but may act as a hub (Fig. 2a), connecting different parts of the network, or one of them may also be a member of a different community (Fig. 2b) to which he or she has a stronger dependency than to his or her co-author.

4 Conclusion and Future Work

Our research provides new insights into detecting strong cliques that can be considered cores of more complex structures, such as communities or protein complexes. We have shown that interacting groups of authors in the studied co-author networks can usually contain at least one pair of authors who are not closely related topologically. This is due to the combination of weights in the network and the neighborhood of these two individuals. A dependency threshold is then used to remove such connections from the network. After applying the dependency threshold, we found that with respect to the dependency threshold,

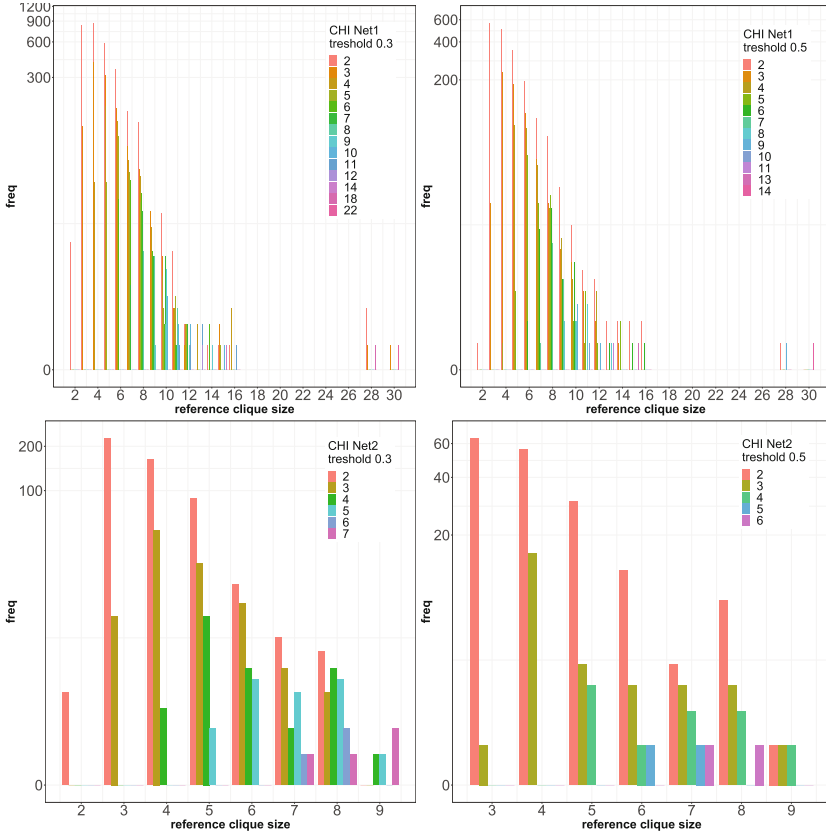


Fig. 6. Sub-clique size distribution after applying the dependency thresholds for CHI Net1 and Net2.

the most frequently related authors are triads and dyads. However, apart from the number and size of handles for both dependencies of 0.3 and 0.5, the trends look similar. We acknowledge that our estimated initial dependency threshold may not be perfect and still requires fine-tuning, as we only performed experiments on two co-authored networks. Distributions of cliques such as visualized in Figs. 3, 4, or 5 could be helpful when we need to decide how to adjust the threshold value.

Finding the optimal threshold value for the dependency poses a challenge for future research, this challenge also opens up additional challenges, such as studying the effect of neighborhood and interdependence thresholds on the properties of detected strong cliques and finding a way to extend the cores with a periphery for discovering structures such as communities, not only in co-authorship networks but ideally in any network as each network may be different, and the required underlying structures may not always consist of cliques.

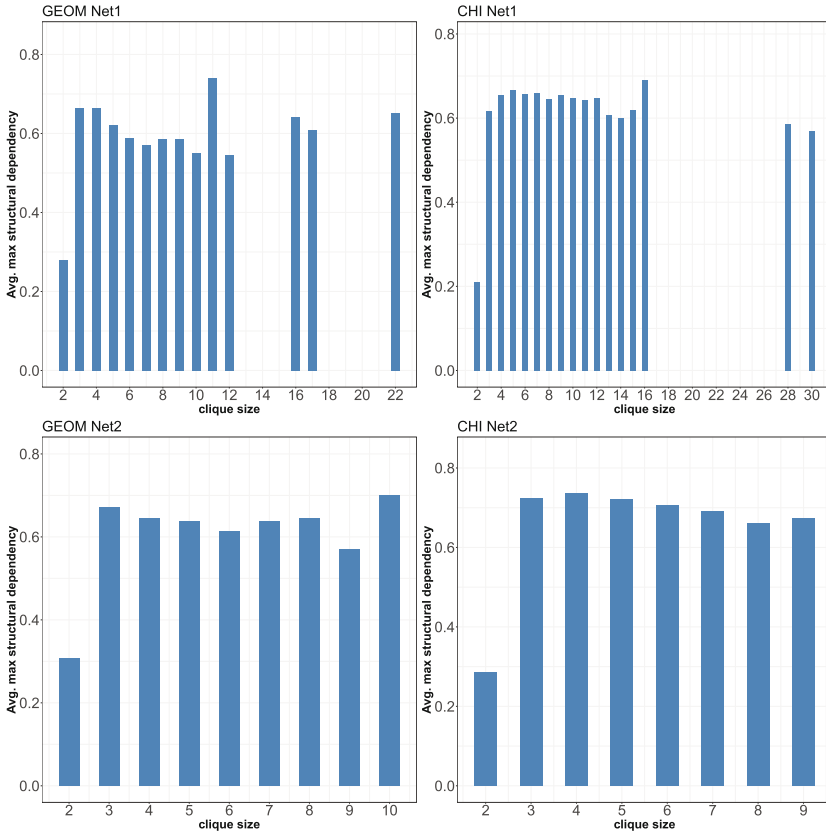


Fig. 7. Average maximum dependency based on clique size

Funding. This work is supported by SGS, VSB-Technical University of Ostrava, under the grant no. SP2023/076, and by Ministry of Health of the Czech Republic under grants no. NU20-06-00269 and NU21-06-00370.

References

1. Batagelj, V., Zaveršnik, M.: Fast algorithms for determining (generalized) core groups in social networks. *Adv. Data Anal. Classif.* **5**(2), 129–145 (2011)
2. Bron, C., Kerbosch, J.: Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM* **16**(9), 575–577 (1973)
3. Chuan, P.M., Son, L.H., Ali, M., Khang, T.D., Huong, L.T., Dey, N.: Link prediction in co-authorship networks based on hybrid content similarity metric. *Appl. Intell.* **48**, 2470–2486 (2018)
4. Csermely, P., London, A., Wu, L.Y., Uzzi, B.: Structure and dynamics of core/periphery networks. *J. Complex Netw.* **1**(2), 93–123 (2013)

5. Gansner, E.R., Hu, Y., Kobourov, S.: GMap: visualizing graphs and clusters as maps. In: 2010 IEEE Pacific Visualization Symposium (PacificVis), pp. 201–208. IEEE (2010)
6. Grodzinski, N., Grodzinski, B., Davies, B.M.: Can co-authorship networks be used to predict author research impact? a machine-learning based analysis within the field of degenerative cervical myelopathy research. *Plos one* **16**(9), e0256, 997 (2021)
7. Halim, Z., Waqas, M., Baig, A.R., Rashid, A.: Efficient clustering of large uncertain graphs using neighborhood information. *Int. J. Approximate Reason.* **90**, 274–291 (2017)
8. Halim, Z., Waqas, M., Hussain, S.F.: Clustering large probabilistic graphs using multi-population evolutionary algorithm. *Inf. Sci.* **317**, 78–95 (2015)
9. Jain, B., Obermayer, K.: Extending bron kerbosch for solving the maximum weight clique problem. arXiv preprint [arXiv:1101.1266](https://arxiv.org/abs/1101.1266) (2011)
10. Kudelka, M., Ochodkova, E., Zehnalova, S., Plesnik, J.: Ego-zones: non-symmetric dependencies reveal network groups with large and dense overlaps. *Appl. Netw. Sci.* **4**(1), 1–49 (2019)
11. Kumar, S.: Co-authorship networks: a review of the literature. *Aslib J. Inf. Manag.* **67**(1), 55–73 (2015)
12. Lambiotte, R., Panzarasa, P.: Communities, knowledge creation, and information diffusion. *J. Inform.* **3**(3), 180–190 (2009)
13. Lü, L., Zhou, T.: Role of weak ties in link prediction of complex networks. In: Proceedings of the 1st ACM International Workshop on Complex Networks Meet Information & Knowledge Management, pp. 55–58 (2009)
14. Luo, F., Li, B., Wan, X.F., Scheuermann, R.H.: Core and periphery structures in protein interaction networks. In: BMC Bioinformatics, vol. 10, pp. 1–11. BioMed Central (2009)
15. Newman, M.: *Networks*. Oxford University Press, Oxford (2018)
16. Newman, M.E.: Coauthorship networks and patterns of scientific collaboration. In: Proceedings of the National Academy of Sciences, vol. 101(suppl.1), pp. 5200–5205 (2004)
17. Newman, M.E.: Who is the best connected scientist? a study of scientific coauthorship networks. In: *Complex networks*, pp. 337–370. Springer (2004). https://doi.org/10.1007/978-3-540-44485-5_16
18. Uddin, S., Hossain, L., Abbasi, A., Rasmussen, K.: Trend and efficiency analysis of co-authorship network. *Scientometrics* **90**(2), 687–699 (2012)
19. Wasserman, S., Faust, K.: *Social network analysis: methods and applications* (1994)



Mosaic Benchmark Networks: Modular Link Streams for Testing Dynamic Community Detection Algorithms

Yasaman Asgari^{1,3,4}(✉), Remy Cazabet², and Pierre Borgnat¹

¹ Univ de Lyon, ENS de Lyon, CNRS, Laboratoire de Physique, 69342 Lyon, France

² Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, UMR5205,
69622 Villeurbanne, France

³ University of Zurich UZH Digital Society Initiative, Rämistrasse 69,
8001 Zürich, Switzerland
yasaman.asgari@math.uzh.ch

⁴ University of Zurich, Institut für Mathematik Winterthurerstrasse 190,
8057 Zürich, Switzerland

Abstract. Community structure is a critical feature of real networks, providing insights into nodes' internal organization. Nowadays, with the availability of highly detailed temporal networks such as link streams, studying community structures becomes more complex due to increased data precision and time sensitivity. Despite numerous algorithms developed in the past decade for dynamic community discovery, assessing their performance on link streams remains a challenge. Synthetic benchmark graphs are a well-accepted approach for evaluating static community detection algorithms. Additionally, there have been some proposals for slowly evolving communities in low-resolution temporal networks like snapshots. Nevertheless, this approach is not yet suitable for link streams. To bridge this gap, we introduce a novel framework that generates synthetic modular link streams with predefined communities. Subsequently, we evaluate established dynamic community detection methods to uncover limitations that may not be evident in snapshots with slowly evolving communities. While no method emerges as a clear winner, we observe notable differences among them.

Keywords: Temporal networks · Dynamic community detection · Network generator

1 Introduction

Community structure is a common feature in real networks. Essentially, a community represents a network pattern where nodes have strong connections within the community and weaker connections with nodes outside[1]. Network science initially emerged when real-world temporal network data, which captures the changing structure of networks over time, was scarce. Consequently, early research on community detection primarily focused on static networks [2]. With

the increasing availability of low-resolution temporal data, such as network snapshots, attention naturally shifted to the dynamic community detection[3]. Nowadays, we have access to highly detailed temporal networks like link streams, making the study of community structures more intricate due to the increased data granularity and time-sensitive nature.

One approach to quantitatively evaluate community detection algorithms is to employ synthetic graph generators to assess their performance and accuracy against a reference ground truth. Recently, several synthetic temporal network generators have been established to simulate snapshots with “slowly evolving” communities, i.e., having a meaningful community structure at each discrete time step. However, link streams fail to meet this criterion. Within a link stream, the frequency of interactions per node per unit of time is exceedingly low. Consequently, during a specific time instance, we can only discern a minimal number of connections, and these connections do not reveal clear community structures. Although we can employ varying or fixed window sizes to segment our network into multiple time slices to achieve low-resolution temporal networks like snapshots, a new line of research has emerged, aiming to develop algorithms tailored to the continuous-timed nature of instantaneous edges [4]. Motivated by this challenge, we intend to address the issue of the absence of a benchmark network for simulating modular link streams. To address this need, we introduce the “Mosaic benchmark network.” Mosaic is designed to provide a reliable framework for evaluating and benchmarking dynamic community detection algorithms. It generates modular temporal networks with continuous-time edges built upon randomly planted partition networks, creating adjustable ground truths.

The paper’s organization is outlined as follows: It begins by briefly reviewing the current literature concerning benchmarks used for creating modular networks. Moving on to Sect. 3, a mathematical framework for link streams and the definition of a temporal community adapted for the context are presented. Then, we explain a comprehensive framework called “Mosaic” for creating link streams with communities. Finally, Sect. 4 involves the application of some dynamic community detection algorithms to the benchmark, aiming to uncover their capabilities and limitations.

2 Related Works

Assessing and comparing community detection algorithms presents a significant challenge. Although real-world datasets can offer valuable insights, it has been shown that node metadata are not the same as ground truth and that treating them as such induces severe theoretical and practical problems [5].

To overcome this limitation, researchers have developed benchmarks to generate synthetic networks for examining algorithm behavior on networks with diverse predefined properties[6]. Synthetic network Benchmarks enable checking an algorithm against:

- *Definition* of communities: since there is no universal definition of community, a benchmark with its ground truth defines what we want to find and check if the method recognizes it.

- *Stability*: the efficiency of a community detection method can be evaluated by testing it on numerous network instances that share similar characteristics.
- *Scalability*: gradually increasing the network size makes it possible to determine how well the algorithm handles larger and more complex networks.

Numerous network benchmarks have been introduced to establish modular static networks. Stochastic Block Models (SBM, also random planted partition graphs) [7] generate networks where edges between nodes in and within communities depend on a provided probability matrix.

As the need for synthetic temporal networks increased, several methods have been outlined in the literature to generate benchmark graphs for evolving communities. An evolving community scenario is defined as a structure characterized by fundamental events for communities such as birth, death, merging, splitting, expansion, contraction, iterative continuation, and the Ship of Theseus, as illustrated and described in [3].

Temporal network benchmarks have been developed with diverse perspectives and aims, yet they all share a common trait: they generate snapshots that reveal clear community structures. For instance, Granell et al. [8] propose two cyclic scenarios (migration and merge-split), and in each snapshot, communities are defined using SBMs. Bazzi et al. [9] introduce a method for generating multilayer networks with community structures by incorporating an SBM with additional interlayer dependency tensors. However, Cazabet et al. [10] argue that utilizing an SBM independently for generating edges in each snapshot is impractical. Therefore, they have developed a method that allows for evolving structures while maintaining the stability of most edges from one time step to the next.

Differing from all preceding approaches, our proposed benchmark introduces a framework that accomplishes two crucial objectives: 1) It enables the representation of novel scenario description generators that do not necessitate the inclusion of progressively evolving structures, and 2) It facilitates the generation of continuous-timed instantaneous edges while maintaining a low computational cost.

3 Mathematical Framework

3.1 Link Stream

Link streams, the category of temporal networks examined in this study, can be perceived as a collection of vertices denoted by V , which engage with one another at specific instances, and the duration of these engagements is considered negligible.

Based on this definition, we can formulate a link stream mathematically as:

Definition 1. *Link Stream: A link stream L is defined as a triple (V, E, T) , where V represents the set of nodes involved in interactions within a defined time domain, $T = [T_s, T_e) \subseteq \mathbb{R}$, and $E \subseteq V \times V \times T$ is the set of temporal edges. Each temporal edge, $l = (u, v, t) \in E$ signifies an instantaneous interaction that took place between node $u \in V$ and node $v \in V$ at time $t \in T$ [4].*

The illustration in Fig. 1 presents a link stream featuring a set of vertices $V = \{v_1, v_2, v_3, v_4\}$ where multiple temporal edges are observed. For example, nodes v_1 and v_4 establish a connection twice within the given time domain, $T = [0, 10)$.

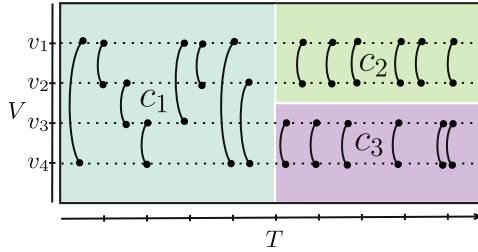


Fig. 1. Modular Link stream: A link stream (V, E, T) is shown, containing 4 nodes $V = \{v_1, v_2, v_3, v_4\}$ interacting several times with each other within the time domain $T = [0, 10)$. A mosaic partitioning, $\mathcal{C} = \{c_1, c_2, c_3\}$ is also observed. This partitioning covers $\{v_1, v_2, v_3, v_4\} \times [0, 10)$ without any overlap.

3.2 Mosaic: A Definition for a Community in Link Streams

Defining a community in a link stream is a challenging task due to the fine-resolution temporal dimension involved. However, any new definition must align with the intuitive understanding of real-world applications. To give an intuition of the meaning of communities in this setting to unfamiliar readers, we illustrate it using a well-known link-stream dataset.

Sociopatterns¹ is a renowned database for real-world link streams acquired in various contexts since 2008 [11]. In these experiments, RFID sensors track real-time proximity, creating co-presence networks between individuals. For example, a substantial dataset comes from a primary school study where 230 pupils and 10 teachers wore sensors for two consecutive days. This study recorded 125,000 face-to-face interactions over 32 h, with a temporal resolution of 20 s. Previous research often viewed its communities as evolving structures influenced by node and edge additions/deletions, identifying them by segmenting time into different slices with various window sizes and then applying dynamic community detection techniques.

In this study, we can observe communities emerging during specific timeframes from students interacting. For instance, students and teachers interact during lecture hours within their respective classes. During lunchtime, students with stronger friendships tend to dine together. An empty community exists at night, indicating no interactions occur during that period.

¹ <http://www.sociopatterns.org/>.

This perspective can be extended to other contexts like people discussing a particular topic on social networks, company meetings, or sports players participating in a match together. Building on this idea, we introduce a new definition called “Mosaic,” which is a straightforward adaptation of non-overlapping communities from static networks to link streams.

A “Mosaic” community is defined as follows:

Definition 2. A Mosaic, denoted as c , is defined as a pair of (nodes, period): $c = (V_c, T_c)$. V_c is set of n nodes denoted as $\{v_1, v_2, \dots, v_n\}$. $T_c \subset \mathbb{R}$ is an time interval, $T_c = [T_{cs}, T_{ce})$ where, T_{cs} and T_{ce} represent the start and end times of a Mosaic c , respectively. It represents the interval in which nodes V are considered part of the community c .

According to this definition, each node is assigned to only one community at any given time, and these communities collectively cover the entire link stream; refer to Fig. 1 for an example. In cases where nodes do not interact significantly for a certain duration, they are assigned to an “empty community”, c_* . Therefore, a Mosaic partitioning can be defined as follows:

Definition 3. Mosaic partitioning: Given a link stream $L = (V, E, T)$, \mathcal{C} is a partitioning containing k mosaics, $\{c_1, c_2, \dots, c_k, c_*\}$, that cover the link stream fully without any overlap. This requirement can be written as follows:

$$\bigcup_{c \in \mathcal{C}} V_c \times T_c = V_L \times T_L \quad \bigcap_{c \in \mathcal{C}} V_c \times T_c = \emptyset$$

The mosaic c_* stands for an empty community.

3.3 Mosaic Link Stream Benchmark

Now, we delve into discussing the Mosaic Benchmark. The proposed framework follows a straightforward five-step process, as depicted in Fig. 2. The whole procedure is implemented as a user-friendly Python library².

² <https://pypi.org/project/mosaic-benchmark/>.

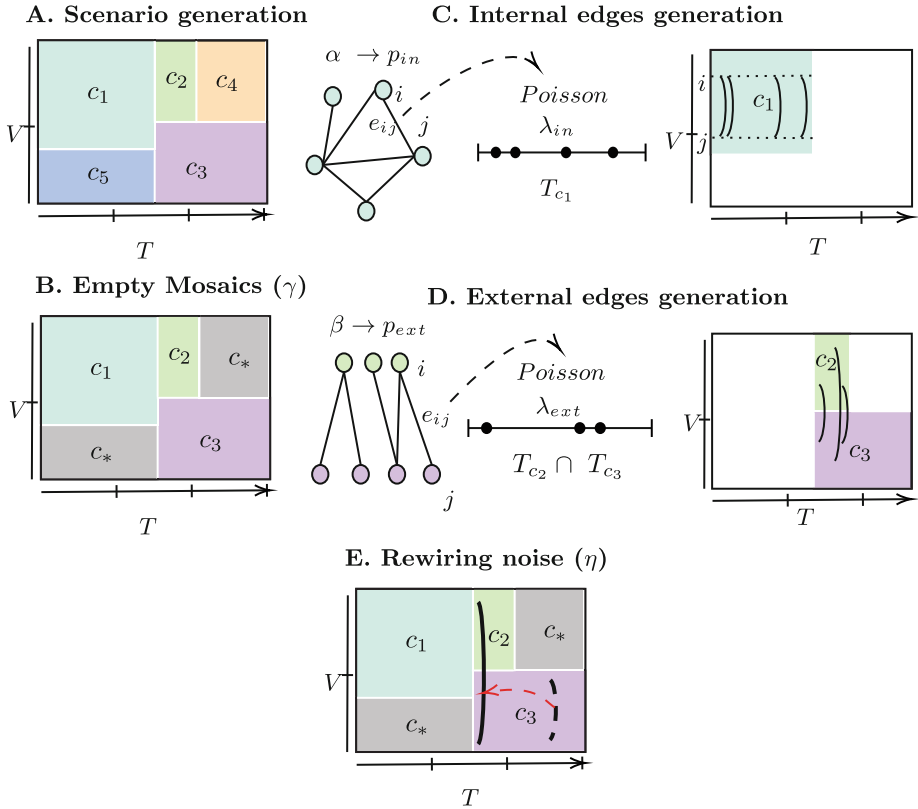


Fig. 2. Random Mosaic Link Stream Benchmark: This figure illustrates a five-step process. Step A involves generating a scenario. Step B focuses on removing mosaics with a probability of γ to create an empty community named c_* . Steps C and D add internal and external edges, respectively. Finally, in the last step, each edge in the link stream can be rewired with a probability of η .

We can condense these five steps into two primary stages:

1. Scenario description: The user describes communities using a scenario, either ad-hoc or generated from a provided scenario generator (Step A). Additionally, some communities may be emptied to match real-world properties (Step B).
2. Edge generation: Initially, edges are formed within the communities (Step C), followed by establishing connections between communities with overlapping timeframes (Step D). Some edges may be rewired to introduce imperfections into the community structures (Step E).

3.4 Scenario Description

In the proposed framework, we use Mosaic partitioning to generate modular link streams according to a scenario. Mosaic partitioning consists of multiple com-

munities in which nodes interact within and between them. It can be generated using three proposed scenario generators in our Python library: Experimental, Snapshots, and Random. A visualization can be found in Fig. 3.

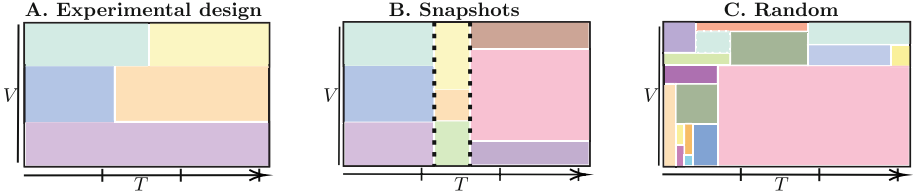


Fig. 3. Mosaic Partitioning Generation This figure illustrates three different approaches for generating Mosaic Partitioning in the Benchmark. **Type A:** User-defined nodes and periods create the desired scenario. **Type B:** The time domain is divided into multiple frames or snapshots, and node sets are randomly assigned to communities within each frame. **Type C:** Communities' time interval and node-set sizes are distributed inhomogeneously, covering the entire link stream.

Experimental: Using an experimental setup, we can intentionally generate various Mosaic communities with specific node counts and time intervals. This enables us to evaluate the performance and precision of a community detection algorithm under straightforward scenarios, gradually increasing complexity to gain insights into the algorithm's capabilities and constraints.

Snapshots: This scenario generator simulates low-resolution temporal networks resembling snapshots. It partitions the time domain into k segments defined by either fixed or variable window sizes. In the fixed case, each segment has a size of $\lfloor \frac{T}{k} \rfloor$, while in the varying case, the time domain is randomly divided. Then, we establish static communities for each segment, ensuring they consist of at least two or more nodes.

Algorithm 1. Snapshot Scenario Generator

- 1: **procedure** SNAPSHOT(T, V, k, Mode)
 - 2: Create an empty list \mathcal{C}
 - 3: Divide the time domain into k segments with fixed or varying window sizes and name it S
 - 4: **for** each s in S **do**
 - 5: Distribute the nodes V randomly into static communities and name the collection U
 - 6: **for** each u in U **do**
 - 7: Add (u, s) to \mathcal{C}
 - 8: **return** \mathcal{C}
-

Random: Generating random scenarios leading to mosaics with different node counts and time intervals is essential for comprehending the capabilities and constraints of a dynamic community detection algorithm. To accomplish this objective, we will suggest a simple procedure for constructing random communities using k iterations in a recursive algorithm. Once we have obtained the algorithm's output, we will proceed to remove communities consisting of just one node or communities that exist for less than a predefined time interval.

Algorithm 2. Random Scenario Generator

```

1: procedure RANDOM( $T, V, k$ )
2:   Create a Mosaic  $M = (T, V)$ 
3:   for  $i$  in 0 to  $k$  do
4:     if  $M$ .sub-mosaics is empty then
5:       Divide  $M$  into 4 sub-mosaics and store them in  $M$ .sub-mosaics
6:     else
7:       Select one sub-mosaics from  $M$ .sub-mosaics and divide it into 4 sub-
mosaics
8:     Flatten the Mosaic  $M$  into  $\mathcal{C}$ 
9:   return  $\mathcal{C}$ 

```

Emptying Mosaics. We consider that Mosaics are assigned to an empty Mosaic c_* with a probability of γ . We mean that within this empty Mosaic, no edges can be active that originate from either inside or outside, affecting the nodes contained within it.

3.5 Generating Edges

This part focuses on generating edges between nodes within and across different communities. We will follow two steps: creating a Backbone connectivity network to establish static connections and using the Poisson Point Process to add a temporal dimension above it; refer to Algorithm 3.

Algorithm 3. Edges Generation

```

1: procedure EDGESGENERATION( $\mathcal{C}, \alpha, \lambda, \beta$ )
2:   Create an empty list  $E$ 
3:   for  $c$  in  $\mathcal{C} \setminus c^*$  do ▷ Generate internal edges
4:      $p_{in}^c = (|V_c| - 1)^{\alpha-1}$ 
5:     List  $S = \mathbf{BackboneConnectivity}(c, c, p_{in}^c)$ 
6:     for  $e$  in  $S$  do
7:       Add PoissonProcessEdge( $e, P_c, \lambda^{cc}$ ) to  $E$ 
8:   for  $(c, c')$  in  $\binom{\mathcal{C} \setminus c^*}{2}$  do ▷ Generate external edges
9:      $p_{ext}^{cc'} = \beta(|V_c| + |V_{c'}| - 1)^{\alpha-1}$ 
10:    List  $S = \mathbf{BackboneConnectivity}(c, c', p_{ext}^{cc'})$ 
11:    for  $e$  in  $S$  do
12:      Add PoissonProcessEdge( $e, P_c \cap P_{c'}, \lambda^{cc'}$ ) to  $E$ 
13:   return  $E$ 

```

Backbone Connectivity Network. This Benchmark assumes that the connectivity between nodes, whether through internal or external edges, remains stable throughout the specified period. This is why we refer to it as the backbone connectivity network. A backbone connectivity network with a parameter p is a random graph in which each edge is present with probability p , independent of others.

We would like to emphasize that for establishing a well-defined internal structure of a community, it is necessary to utilize an appropriate range of values for p . This range's selection should depend on the number of vertices within the community. To achieve this, we will adopt the model described in [10], which provides the formula for p_{in}^c as follows:

$$p_{in}^c = (|V_c| - 1)^{\alpha-1}$$

Here, $\alpha \in (0, 1]$ is a hyperparameter named community density coefficient shared between communities. When the value of α is increased, the probability of p_{in}^c also increases, leading to denser clusters. If α is set to 1, each community in Mosaic becomes a clique.

The external probability between two communities c and c' denoted as $p_{ext}^{cc'}$ is defined as:

$$p_{ext}^{cc'} = \beta(|V_c| + |V_{c'}| - 1)^{\alpha-1}$$

This hyperparameter $\beta \in [0, 1]$ is related to “community identifiability,” which is shared among all communities. Increasing the value of β results in more external edges between communities, making it more challenging for algorithms to identify each community as a separate cluster. In other words, β controls the external density of backbone connectivity by treating two communities as a single entity.

Poisson Point Process. To simplify the analysis, we assumed that the edges in a given backbone connectivity network follow a memory-less stochastic process

for their activation times. For each edge $e = (i, j)$ in the backbone connectivity network, we generate an independent and identically distributed random Poisson point process with a rate parameter $|T|\lambda$. This rate parameter determines the average number of this edge active times within the time frame T . Then, we use the uniform distribution to distribute this number of occurrences in the selected period. This means the edge time arrivals are uniformly spread over the interval T [12]. To establish connections within a community, we set T equal to T_c and $\lambda = \lambda_{in}^c$. Conversely, when creating the connections between two different communities c and c' , T is defined by the overlap between T_c and $T_{c'}$. Furthermore, to generate external edges between communities c and c' , we utilize a coefficient $\lambda_{ext}^{cc'}$.

Combining these, to create the temporal dimension, we need a symmetric matrix λ of size $k \times k$, where k represents the number of communities. The main diagonal of this matrix will be utilized for generating internal edges, and non-diagonal elements can be employed for external edges if there is a non-empty time overlap ($T_c \cap T_{c'} \neq \emptyset$) between the communities c and c' .

In both the step of generating backbone connectivity networks and adding a Poisson point process layer, each community is handled independently, which can be efficiently parallelized. This enables handling large networks in a reasonable time. Additionally, finding an upper bound for memory and time complexity can not be provided due to dependence both on time and structure.

Rewiring Noise. A prior study [13] proposed that it is possible to distinguish a stable core within communities from random, short-lived fluctuations in real temporal networks. In our Benchmark, edges go through a rewiring process with a probability of $\eta = [0, 1]$ to highlight imperfections in community structures. During this step, for edge (u, v, t) selected for rewiring, we randomly choose two communities $c \neq c_*$ and $c' \neq c_*$ with non-empty time intersection. We then select two nodes $u \in V_c$ and $v \in V_{c'}$ where $u \neq v$, with a timestamp t randomly chosen within the $T_c \cap T_{c'}$ time frame.

4 Experiments

In this section, we use different community detection algorithms on an instance of our framework to assess their performance in identifying communities compared to the ground truth. Among the many dynamic community detection algorithms available, we have selected four that operate by aggregating the link stream into snapshots using window sizes. These algorithms are previously implemented in “tnetwork” Python package [14], as detailed in the existing literature review [3].

The algorithms compared in this paper are the following:

- **No-Smoothing:** the approach involves applying a static algorithm (in this case, the Louvain method) to each snapshot. Then, the most similar communities in consecutive steps are matched based on the Jaccard Coefficient with a coefficient named θ , set to 0.3 here.

- **Implicit-Global:** in this method, the Louvain algorithm is executed at each snapshot, but instead of initializing it with each node in its own community, the previous partition is utilized as the seed.
- **Label Smoothing:** this method first identifies communities in each slice. Then, attempts are made to match communities across different snapshots, forming a survival graph. A community detection algorithm is applied to this survival graph, resulting in dynamic snapshot communities.
- **Smoothed-Graph:** in this approach, the Louvain method is run at each slice t on a graph with a smoothed adjacency matrix defined as follows:

$$A_{ij}^t = \alpha \cdot A_{ij}^t + (1 - \alpha) \cdot C_{ij}^{t-1},$$

where $C_{ij}^{t-1} = 1$ if nodes i and j belong to the same community at step $t - 1$, and 0 otherwise.

We considered a link stream L consisting of 100 nodes $|V| = 100$ interacting over the time interval $T = [0, 100)$. Then, we created a scenario using a Random scenario generator (as detailed in 3.4) with an iteration parameter $k = 30$. Additionally, we applied a probability $\mu = 0.2$ to empty Mosaic communities, resulting in $|\mathcal{C}| = 61$ communities with an average node count of $\bar{V}_c = 11.7$ and an average time interval of $\bar{T}_c = 10.2$, excluding the empty community c_* .

For the edge generation phase, we fine-tuned the parameters $\alpha = 0.9$ and $\beta = 0.1$. A high α value signifies strong connections among nodes within the internal backbone connectivity networks, while a low β value indicates sparse connections between communities in the external backbone connectivity networks. This configuration results in clear and distinguishable communities regarding the backbone structure.

To further emphasize this characteristic, we maintained the same values for $\lambda_{in} = 0.4$ and $\lambda_{ext} = 0.1$ for all communities, signifying that, in the Poisson Point Process, the likelihood of an edge forming within a community is higher than it is for an external one. It resulted in 28365 edges. Additionally, we exclude the rewiring process for temporal edges.

Since the chosen dynamic community detection methods are not tailored for link stream cases and are designed for low-resolution temporal networks, we convert our sample modular link stream to snapshots with a relatively small fixed window size 2. Subsequently, we applied community detection algorithms to the aggregated snapshots and extracted these communities for the purpose of comparison and visualization.

As depicted in Fig. 4, it is evident that different algorithms can yield varying interpretations of a benchmark network instance. Assessing the accuracy of community detection algorithms under such circumstances can pose a significant challenge. Similarly, when employing aggregation techniques on a link stream, diverse interpretations may arise. In certain social and political contexts, these variations can be contentious and potentially lead to misleading conclusions.

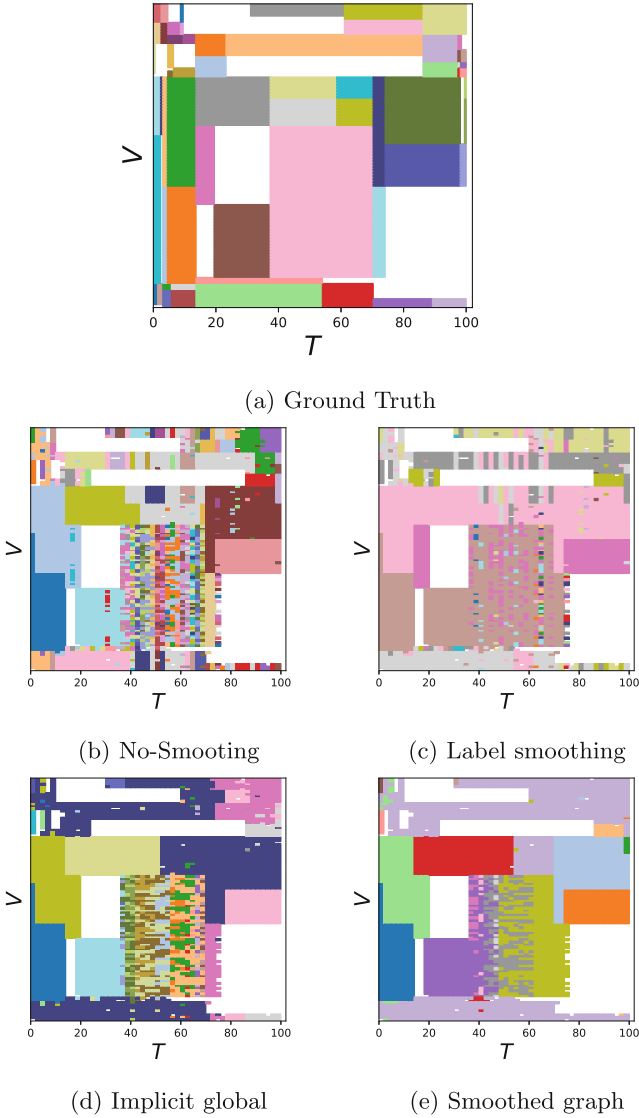


Fig. 4. Experiments: Comparison of partitions obtained using all methods on a sample Random scenario

To conduct a more rigorous experiment, we introduce a new parameter denoted as $\phi = 1 - \alpha = \beta$. A higher ϕ parameter value results in communities being less identifiable. We varied the ϕ value from 0 to 0.5 with the step of 0.1, did the same generation process, and aggregated the link streams to snapshots using a window size of 2, then identified communities ten times with the algorithms.

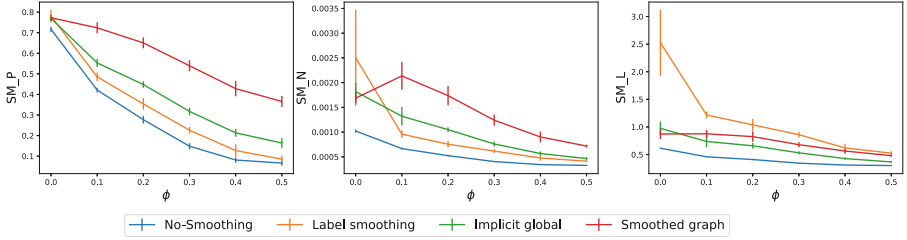


Fig. 5. Experiments: Comparison of smoothness measures obtained using all methods on a sample of a Random scenario.

Smoothness values were determined utilizing the formulas for smoothness presented in the work by Cazabet et al. [10], namely SM-P, SM-N, and SM-L. In all these smoothness metrics, a higher value indicates superior performance. The obtained values were averaged for each ϕ .

As depicted in Fig. 5, in terms of smoothness, two methods have high scores for the three aspects: Implicit-Global and Smoothed-Graph. Label-smoothing has the highest scores in most settings for the SM-L scores, which measure label smoothness. No Smoothing is the least stable in most cases. These results support the findings of other benchmarks presented in [10].

5 Discussion and Conclusions

In summary, we introduced the Mosaic benchmark networks as a new framework for generating modular link streams. These temporal networks enable the evaluation of dynamic community detection algorithms in terms of accuracy and performance. Additionally, we can provide a quantitative assessment based on community definition, stability, and scalability by simulating adaptable ground truth to determine an algorithm’s capabilities and constraints. Furthermore, our framework acts as a foundational platform for assessing algorithms tailored for link streams, especially in the later stages of research in this domain.

The time complexity of this framework can benefit from parallelization because it avoids SBM calculations for each small step, and the future does not depend on past or present data. Moreover, when simulating link streams with millions of edges, it is crucial to efficiently organize edge and node storage. Multi-layered hash functions can enhance memory allocation for storing network edges instead of a large edge stream. To create directed modular link streams, we propose using the benchmark network with an asymmetric Poisson rate matrix (λ), along with two additional parameters, α_{out} and β_{out} for edge generation phase. However, it is essential to carefully check if created communities match the definition of communities in real-world directed link streams.

We applied our framework to explore differences among communities identified by various dynamic community detection algorithms. We emphasize that the aggregation method using window sizes has its limitations, and there is a

crucial need for a more comprehensive investigation into effectively handling continuously timed edges rather than aggregating them.

In conclusion, it is important to explore the characteristics of communities formed through the provided scenario generators. This framework incorporates various parameters such as γ for emptying communities, and (α, β, λ) for edge generation. Furthermore, the parameter η plays a crucial role in evaluating an algorithm's robustness. We advocate for further research to enhance the comprehension of dynamic community algorithms, achieved through the fine-tuning of these parameters.

Funding. We thank the University of Zurich and the Digital Society Initiative for (partially) financing this project conducted by Yasaman Asgari. The work has been supported by the ANR grant DARLING ANR-19-CE48-0002, the ANR grant BITU-NAM ANR-18-CE23-0004, and the CHIST-ERA grant CHIST-ERA-19-XAI-006, for the GRAPHNEX ANR-21-CHR4-0009 project. We also thank Victor Brabant for his valuable discussions.

References

1. Girvan, M., Newman, M.E.: Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99**(12), 7821–7826 (2002)
2. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3–5), 75–174 (2010)
3. Rossetti, G., Cazabet, R.: Community discovery in dynamic networks: a survey. *ACM Comput. Surv. (CSUR)* **51**(2), 1–37 (2018)
4. Latapy, M., Viard, T., Magnien, C.: Stream graphs and link streams for the modeling of interactions over time. *Soc. Netw. Anal. Min.* **8**, 1–29 (2018)
5. Peel, L., Larremore, D.B., Clauset, A.: The ground truth about metadata and community detection in networks. *Sci. Adv.* **3**(5), e1602548 (2017)
6. Lancichinetti, A., Fortunato, S.: Community detection algorithms: a comparative analysis. *Phys. Rev. E* **80**(5), 056117 (2009)
7. Holland, P.W., Laskey, K.B., Leinhardt, S.: Stochastic blockmodels: first steps. *Social networks* **5**(2), 109–137 (1983)
8. Granell, C., Darst, R.K., Arenas, A., Fortunato, S., Gómez, S.: Benchmark model to assess community structure in evolving networks. *Phys. Rev. E* **92**(1), 012805 (2015)
9. Bazzi, M., Jeub, L.G., Arenas, A., Howison, S.D., Porter, M.A.: Generative benchmark models for mesoscale structure in multilayer networks. *arXiv preprint arXiv:1608.06196*, p. 20 (2016)
10. Cazabet, R., Boudebza, S., Rossetti, G.: Evaluating community detection algorithms for progressively evolving graphs. *J. Complex Netw.* **8**(6), cnaa027 (2020)
11. Cattuto, C., Van den Broeck, W., Barrat, A., Colizza, V., Pinton, J., Vespignani, A.: Dynamics of person-to-person interactions from distributed RFID sensor networks. *PLOS ONE* **5**, e11596 (2010)
12. Ross, S.M.: *Introduction to probability models*. Academic Press (2014)
13. Kobayashi, T., Takaguchi, T., Barrat, A.: The structured backbone of temporal social ties. *Nat. Commun.* **10**(1), 220 (2019)
14. Cazabet, R., Boudebza, S., Jorquera, T.: *tnetwork* (2023)



Entropic Detection of Chromatic Community Structures

Franck Delaplace^(✉)

IBISC-lab, Paris-Saclay University, Univ. Evry, Paris, France
franck.delaplace@univ-evry.fr
<https://www.ibisc.univ-evry.fr/~delapla/>

Abstract. The detection of community structure is probably one of the central challenges in complex networks whose objective is to identify internal organizations of people, molecules or processes within a network. The issue is to provide a network partition representative of this organization so that each community presumably gathers nodes sharing a common mission, purpose or property. Usually, this identification is based on the difference in connectivity density between the interior and border of a community. Indeed, nodes sharing a common purpose or property are expected to interact closely. Although this rule appears mostly relevant, some fundamental scientific problems like disease module detection highlight the inability to meaningfully determine the communities by this connectivity rule. The main reason is that the connectivity density may not be correlated to a shared property or purpose. Another paradigm is therefore necessary to properly formalize this problem in order to accurately detect these communities. In this article we propose a new framework to study this novel community formation property. Considering that colors formally represent shared properties, the problem becomes to maximize groups of nodes of the same color within communities. We introduce a new measurement called *chromatic entropy* assessing the quality of the community structure regarding the color constraint. Next we propose a novel algorithm detecting the community structure based on this new community formation paradigm.

Keywords: Community structure · Detection algorithm · Complex Network

1 Introduction

Complex networks model component interactions in diverse real-world domains as in sociology with social or friendships networks, computer science with WEB, and biology with regulatory, metabolic or neural networks. Nodes of these networks are often arranged in closely tight groups called communities. These communities delineate the organizational supports of function, property, purpose or categories. They thus highlight a network structure providing an organizational understanding behind the interactions. Formally, the goal is to identify a node partition of the network based on some grouping rules. A *community structure* is

a partition of the vertices of a graph defined according these rules. Although there is no firm answer concerning these rules [1], it is commonly admitted that the definition of a community relates to a difference in connection density between its interior and its boundary. The density of connection between nodes inside a community must be higher than the density of connection across communities. Such community is called the *topological community* [2]. Community detection algorithms capture this difference of connection density for detecting communities in a network [3,4]. The quality of a community structure is evaluated by a measure assessing this partitioning. A recognized standard is the *modularity* introduced by Newman [5] which is based on the comparison of the network with a random one having the same topological characteristics (*i. e.*, same number of nodes, same node degree). Therefore a good measure must be greater than a community structure having the same characteristics but obtained by chance because this reveals an organizational bias showing an intentional design on the interactions. Finding a community structure maximizing the modularity is NP-hard [6] and different heuristics have been proposed for detecting the optimal community structure [3,7–9].

While the concept of community is central in network science, the connection density rule sometimes fails to identify the meaningful community structure of a network, thus restricting the applicability of community detection algorithms. It is notably the case when the community formation is not correlated to a connection density. *Disease module* characterizes concretely this scenario. A disease module groups genes which are mechanistically linked to the same pathophenotype. The study of the modularity of human disease would provide a causal understanding of the pathogenesis strengthening the etiological explanation and rationally determine clues for drug target discovery.

In [10], the authors carefully demonstrate that disease module are not topological module/community. By using three representative, methodologically distinct algorithms on community structure detection based on density connection, the authors show that the disease genes gathered in a community by connection density method are drastically under-represented, thus prohibiting the ability to assign communities to diseases. Furthermore, they also show that this lack of representativeness is not due to insufficient knowledge about genetic diseases, but rather to the inability of topological communities to properly represent disease modules. This conclusion is also confirmed by other works on disease modules [11–13]. Due to its paramount importance for health, the identification of disease modules clearly indicates the urgent need to extend this framework for detecting such community structures.

More generally, such a situation may occur whenever nodes possessing the same property justifying their membership in the same community are sparsely connected. If only connectivity prevails the algorithm thus fails to bring together nodes of the same property as there is no a priori relationships between the shared properties and connections. The shared common property which is responsible for the community formation must be understood in a broad sense including a wide variety of situations such as involvement in the same process or

function, membership of a social or ethnic group, identical characteristics, sharing a common topic of interest, common purpose or mission etc., more generally any trait that can be shared by a community and qualifying its members.

It thus seems greatly beneficial to extend the methodological framework related to network community detection so that the resolution of this problem addresses a broader context than disease modules. Therefore our objective is to generalize its principles in order to characterize a framework dedicated to community detection which relies on a shared common property.

In this context, a property will be assimilated to a “color” leading to assign the same color to the nodes having the same property. Accordingly, the issue of *chromatic community structure* detection is to find communities of connected nodes that maximize the density of the major color within each while maintaining the connectedness constraint in each community.

In this article, we study the chromatic community structure detection problem and propose an algorithm for finding partition of communities based on color detection. In Sect. 2 we mathematically formalize the problem. We then define in Sect. 3 the *chromaritic entropy* which is a measure assessing the significance of a chromatic community structure. We detail in Sect. 4 an algorithm finding a chromatic community structure before concluding (Sect. 5). An extended version of this article including a benchmark of the algorithm can be found in [14].

2 Formalizing the Coloring

In this section we introduce the basic notions related to graph coloring. Let $G = \langle V, E \rangle$ be a graph where V is a set of vertices and $E \subseteq V \times V$ a set of edges, a *community* p is a subset of V (i. e., $p \subseteq V$) and a *community structure* P is a partition of V . A community structure based on color selection criteria is called a *chromatic community structure*.

Coloring Profile. Coloring assigns a color to each vertex of a graph which is described by a *coloring profile* corresponding to an application from vertex to color $c : V \rightarrow C$ where C denotes the set of colors. The set of colors C will be represented by an integral interval $[1, r]$ where integers define colors. For example $c = \{1 \mapsto 1, 2 \mapsto 3, 3 \mapsto 1, 4 \mapsto 2, 5 \mapsto 1, 6 \mapsto 3\}$ assigns color 1 to nodes 1, 3, 5, color 2 to node 4 and color 3 to nodes 2, 6 (see Fig. 1). The restriction of the coloring to community denoted c_p for community $p \subseteq V$ is defined as: $c_p = \{v \mapsto c(v) \mid v \in p\}$.

If the vertices correspond to an integral interval $V = [1, n]$ then the coloring profile can be described by a vector such that the index stands for a vertex label and its corresponding value for a color (i. e., $c(i) = k \iff i \mapsto k \in c$). For the former example (Fig. 1) the vector representation is (1, 3, 1, 2, 1, 3).

Colored Graph. A *colored graph* is a 3–uple $\langle V, E, c \rangle$. The colored graph in Fig. 1 uses 3 colors $C = [1, 3]$ where: green= 1, red= 2 and yellow= 3. From its coloring profile:

$$c = \{1 \mapsto 1, 2 \mapsto 3, 3 \mapsto 1, 4 \mapsto 2, 5 \mapsto 1, 6 \mapsto 3\},$$

the chromatic community structure is:

$$P = \{p_1 = \{1, 3, 4, 5\}, p_2 = \{2, 6\}\},$$

we deduce the following coloring profiles restricted to p_1, p_2 :

$$c_{p_1} = \{1 \mapsto 1, 2 \mapsto 3, 4 \mapsto 2, 5 \mapsto 1\}, c_{p_2} = \{2 \mapsto 3, 6 \mapsto 3\}.$$

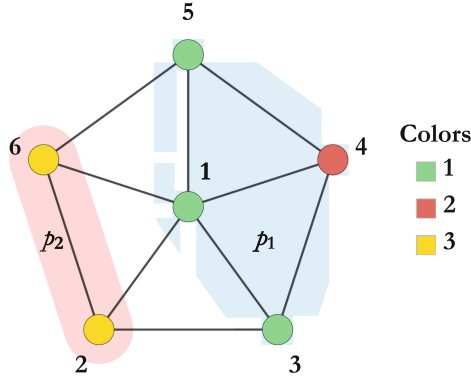


Fig. 1. Community structure of a colored graph.

Chromatic Function. A *chromatic function* $\chi : (V \rightarrow C) \rightarrow C \rightarrow \mathbb{N}$ counts the number of occurrences of each color in a coloring profile. The formal definition of the chromatic function is based on the *counting operator* (**Count**) which is a function counting the positions/nodes of each element corresponding to values of a vector or a function. **Count**(X, y) specifically counts the number of occurrences of element y in vector/function X :

$$\mathbf{Count}(X, y) = \{y \mapsto |\{i \mid X(i) = y\}|\}.$$

$$\mathbf{Count}(X) = \bigcup_{i=1}^{|X|} \mathbf{Count}(X, X(i)).$$

The chromatic function is thus defined from a coloring profile c as:

$$\chi_c = \bigcup_{k \in C} \mathbf{Count}(c, k) \tag{1}$$

The chromatic function of c of the example in Fig. 1 is:

$$\chi_c = \{1 \mapsto 3, 2 \mapsto 1, 3 \mapsto 2\}.$$

Dominant Color. A coloring profile with d vertices of the same color, will be called a d -coloring profile. This notion is also applied to community from their local coloring profile. A d -colorful community p implies that:

$$\exists k \in C : \chi_{c_p}(k) = d. \quad (2)$$

Notice that these coloring profiles may also have several subsets of vertices with the same color of cardinality greater or equal to d . The graph in Fig. 1 is a 3-coloring profile for color 1, but also a 2-coloring profile for color 3, and 1-coloring profile for color 2.

Among the d -coloring profiles we specifically focus on the class of profiles where d is the cardinality of the color occurring the most. These profiles are said d -dominant by this main color. Hence a coloring profile is d -dominant if and only if:

$$\exists k \in C : \chi_c(k) = d \wedge \forall k' \in C : \chi_c(k') \leq d. \quad (3)$$

In this case, color $k \in \arg \max \chi_{c_p}$ is said *dominant*. In Fig. 1 the dominant color is 1 and the coloring profile is thus 3-dominant. By extension, a community is said d -dominant if the restriction of the coloring profile to this community is d -dominant. In Fig. 1, p_1 is 3-dominant for color 1 and p_2 is 2-dominant for color 3. Notice that several dominant colors may exist in a coloring profile.

3 Chromatic Entropy

Although, the significance of the colorful communities closely depends on the application fields for interpreting the colors, the issue is to define a generic measure assessing the significance of a chromatic community structure. Basically this measure is related to the color dominance in community. Intuitively more a color dominates more significant a community is.

However, this characteristic is not enough for relevantly qualifying the significance of community structure. Indeed, as an extreme illustrative example, let us consider a community structure where each community is reduced to a single node. Such structure leads to optimal coloring of the communities since the single node owns the dominant color in its community because it covers it totally. However such community structure clearly tells us nothing of value about community organization since all nodes remain isolated.

A relevant measure should assess the intentionality behind the design of a community. By considering that the human design driven by intention is opposed to chance, a significant community should thus lead to gather more nodes of the dominant color than would be expected by chance. Indeed, the situation that cannot be delivered by chance necessary underlies a mechanistic organization representing a human intention. As a result, we can confidently conclude that the structure of the chromatic community excluding chance would provide a meaningful structure supporting intentional organization. Such perspective raises two major issues: 1) defining a measure characterizing the intention in community design, 2) formally characterizing the probability to randomly generate a d -colorful community.

3.1 Chromatic Entropy Definition

In the analysis of complex systems, entropy is a concept commonly used to quantify disorder, randomness, chaos, or uncertainty in various domains. By integrating entropy into the community detection process, the goal is to find a partition that maximizes the significance of community structure while minimizing randomness within communities considered a sign of disorder and community disorganization. This approach would reveal meaningful communities in complex networks, leading to a better understanding of the underlying community structure and their organizational law.

In our context, the *chromatic entropy* H quantifies the intentionality of the community design. The chromatic entropy will relate to the coloring of a community obtained by chance: the more likely a community is to be colored by chance, greater its entropy. A community structure with a small entropy thus emphasizes a meaningful community structure. Accordingly, the chromatic entropy is based on the quantification of the community organization intentionally designed, called the *intentionality measure* and denoted $I : \Delta_1 \rightarrow \mathbb{R}$. Intuitively, this measure defines how much a community is intentionally organized. It is semantically equivalent to the measure of information introduced by Shannon. It is expected that the higher the probability of random community generation, the lower the Intentionality measure.

Let $\Delta_m = \{(p_1, \dots, p_m) \mid 0 \leq p_i \leq 1 \wedge \sum_{i=1}^m p_i \leq 1\}$ be the sets of (m -ary), possibly incomplete, probability distributions on m communities. The entropy H is a continuous function defined as $H : \Delta^* \rightarrow \mathbb{R}$ where $\Delta^* = \bigcup_{m \geq 1} \Delta_m$. Δ^* is used for mathematical convenience to accommodate any community structure cardinality and $\Delta_1 \subseteq [0, 1]$ stands for a subset of the unit interval.

For characterizing the chromatic entropy, we focus on the axiomatic properties framing the definition of this function (Table 1). Notice that the maximality is a property specific to our context that does not necessarily apply to the other notions of entropy as are the other properties [15]. Beside, the Shannon entropy [16] cannot be straightforwardly used due to the maximality property, since $-p \log_2 p = 0$ with $p = 1$ and not ∞ .

By setting the intentionality measure as $I(p) = \log_2(1 - p)$ which fulfills the expected requirements the chromatic entropy can be finally defined as follows (Definition 4):

$$H(p) = -pI(p) = -p \log_2(1 - p) \quad (4)$$

The extension to a distribution of probabilities Δ_m follows the usual generalized form of entropy-function (Definition 5):

$$H(p_1, \dots, p_m) = \sum_{i=1}^m H(p_i). \quad (5)$$

Table 1. Properties of the entropy

Definition	Property
Non negative : The entropy cannot be negative since it is a metric	$H(p) \geq 0$
Expansibility: adding a community with probability zero does not change the entropy of the structure	$H(p, 0) = H(p)$
Symmetry: The entropy is insensitive to a permutation on probability distribution	$H(p_1, p_2) = H(p_2, p_1)$
Sub Additivity: The entropy of a community structure is less than or equal to the sum of the entropies of the communities composing it	$H(p_1 p_2) \leq H(p_1) + H(p_2)$
Minimality: The community structure is assumed to be totally meaningful with a minimal entropy when the probability is null	$H(0) = 0$
Maximality: The entropy is maximal when the probability is 1 because the community is assumed to be fully random	$H(1) = \infty$

3.2 Probability of Random Coloring

The probability to randomly generate a d -colorful community of n nodes with a particular color chosen among r colors by chance is defined by the ratio of the favorable cases to the possible cases. The number of the whole possible colored communities is r^n corresponding to the cardinal of the complete enumeration of the possible combinations of vertex coloring among r colors. The definition of the favorable cases necessitates to combinatorically enumerate them which is harder to characterize than the possible cases. Two issues are addressed:

1. the enumeration of the d -colorful communities of size n considering r colors;
2. the enumeration of the d -dominant colorful communities of size n considering r colors.

Enumeration of d -colorful communities. Different coloring of d vertices are obtained using any color. Let D_k be the set of colorful communities having d vertices of color k , the count of all communities containing a d -color profile obviously corresponds to the cardinality of the union of these sets, namely: $|\bigcup_{k=1}^r D_k|$. Its enumeration formula is based on the *Poincaré sieve* (inclusion-exclusion principle). Theorem 1 provides the general enumeration formula deduced from the Poincaré sieve.

Theorem 1. *The count of d -colorful communities of size n with r colors is given by κ function:*

$$\kappa(r, n, d) = \sum_{k=1}^{\min(r, \lfloor \frac{n}{d} \rfloor)} \frac{(-1)^{k-1} \binom{r}{k} n! (r-k)^{n-kd}}{(n-kd)! (d!)^k}$$

The proof and more complete explanations are in [14] □

Enumeration of the d -dominant colorful communities The domination implies to include the dominance constraint in comparison to the d -colorful

communities enumeration, leading to specify the different equivalence classes of communities complying with the domination conditions 3. Since the conditions of domination are only based on the number of vertices of the same color regardless the color, if two chromatic functions of two communities p, q are equal up to a permutation on colors $\pi : C \rightarrow C$, $\chi_{c_p} = \pi \circ \chi_{c_q}$ then these communities share the same domination property and thus belong to the same equivalence class related to the color distribution.

We introduce the notion of *chromatic signature* σ to capture this equivalence on chromatic functions. A signature of a chromatic function is a vector of color count corresponding to its ordered image (Definition 6)

$$\sigma_p = \mathbf{Sort} \circ \mathbf{Img} \chi_{c_p} \tag{6}$$

Several chromatic functions may have the same signature. For example the two chromatic functions: $\{1 \mapsto 0, 2 \mapsto 3, 3 \mapsto 2\}$ and $\{1 \mapsto 3, 2 \mapsto 0, 3 \mapsto 2\}$ have the same chromatic signature which is: $(0, 2, 3)$. The signatures are at the heart of the combinatorial formula enumerating the d -dominant coloring profiles by abstracting the chromatic functions. The count of the dominant colorful communities is defined as (Theorem 2):

Theorem 2. *The count of all possible d -dominant communities of size n with r colors is given by γ function:*

$$\gamma(r, n, d) = n!r! \sum_{\sigma \in \mathcal{S}_{r,n,d}} \frac{1}{\prod_{s \in \mathbf{Img} \circ \mathbf{Count}(\sigma)} s! \prod_{i=1}^r \sigma(i)!}$$

The proof and more complete explanations are in [14]. □

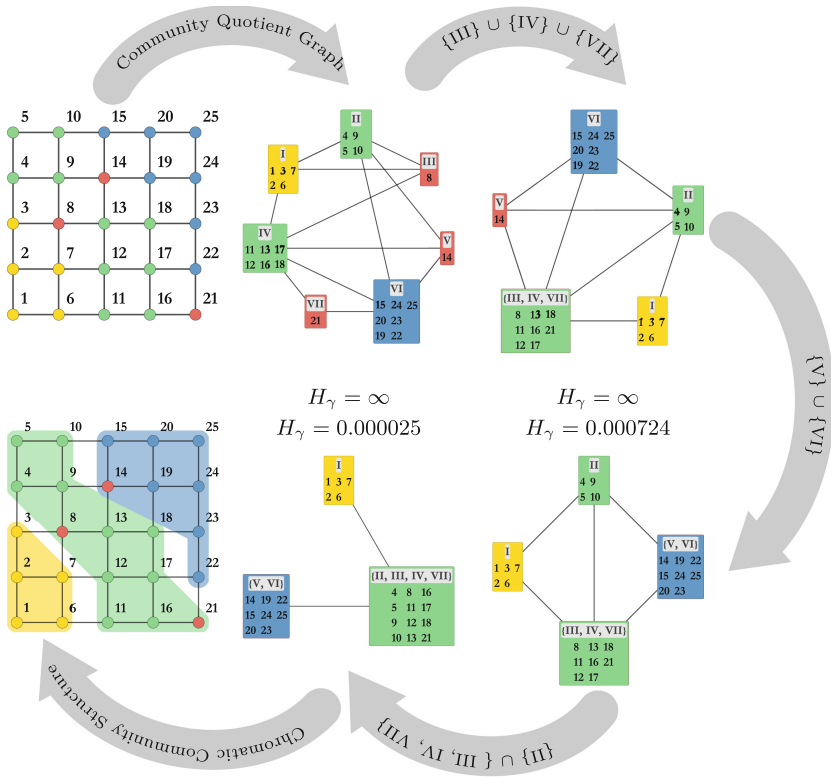
Finally, the probability of random coloring corresponds to the ratio of the favorable cases given by κ or γ to the possible cases given by the number of all possible colorful communities r^n . Therefore, these probabilities are respectively:

$$p_\kappa = \frac{\kappa(r, n, d)}{r^n}, \quad p_\gamma = \frac{\gamma(r, n, d)}{r^n} \tag{7}$$

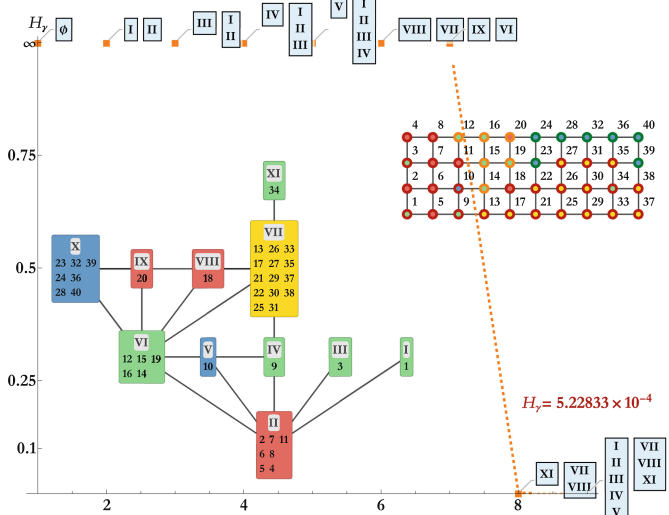
Notice that if the coloring is reduced to 1 color ($r = 1$) or the community is imposed to be a singleton ($n = d = 1$) then $p_\kappa = p_\gamma = 1$ and thus $H_\kappa = H_\gamma = \infty$ which follows the intuition since by considering these restrictions any community obtained by chance is optimal.

4 Chromatic Community Structure Detection

The chromatic community detection algorithm (CHROCODE) finds a partition of a colored graph minimizing the chromatic entropy H . The algorithm is divided in two phases: first a partition grouping connected nodes of the same color is built, forming a partition of monochrome communities, and next these communities are iteratively merged to decrease the chromatic entropy until no merges



— Example with 40 nodes —



The labels of the cluster of nodes that are vertices of the quotient graph are in Roman while the nodes of the original graph are labeled in Arabic.
 PARAMETERS: 1) $n = 25, \delta = 2, r = 4$, 2) $n = 40, \delta = 1, r = 4$

Fig. 2. CHROCODE algorithm steps.

```

function CHROCODE( $G = \langle V, E, c \rangle$  : colored graph,  $\delta$  : radius,  $\omega$  : probability law)
   $P \leftarrow$  partition of  $G$  with monochrome communities.
   $E_P \leftarrow \{(p, p') \mid \exists v \in p, \exists v' \in p' : (v, v') \in E, p, p' \in P\}$ ;  $\triangleright \langle P, E_P \rangle$  quotient graph
   $W \leftarrow P$ ;
  while  $W \neq \emptyset$  do  $\triangleright$  Assemble the communities.
     $p \leftarrow$  community with  $H_\omega$  maximal;
     $W \leftarrow W \setminus \{p\}$ ;
     $N \leftarrow$  neighbors of  $p$  at distance  $\delta$  at most;
     $h \leftarrow H_\omega(G, P)$ ;
     $SP_{\min} \leftarrow$  set of communities from a path in  $N$  minimizing  $H_\omega$  less than  $h$  ;
    if  $SP_{\min}$  is not empty then  $\triangleright$  Community update
       $p = \bigcup_{p_i \in SP_{\min}} p_i$ ;  $\triangleright$  Merge the communities of the path;
       $P \leftarrow (P \setminus SP_{\min}) \cup \{p\}$ ;
       $E_P \leftarrow$  update the quotient graph;
       $W \leftarrow P$ ;
    end if
  end while
  return  $P$ ;
end function

```

Algorithm 1: Chromatic Community Structure Detection Algorithm

can improve the solution. At each step a node p with the maximal entropy is selected and merged with neighbor q for minimizing at most current the chromatic entropy. The node-communities located in the shortest path from p to q are also merged in order to fulfill the connectedness property within the new resulting community. Once the assembly of nodes is achieved they will now form a new community-node corresponding to their union. The quotient graph is then updated by replacing the merged nodes by this new node-community. The process ends when no merges decrease the entropy of the current community structure. The outline of the algorithm is given in Algorithm 1. Figure 2 shows the evolution steps of the algorithm.

CHROCODE is freely distributed in two open-source implementations Mathematica [17] and Python [18].

Let $\langle G, E, c \rangle$ be a colored graph, the complexity of the first phase is in $\mathcal{O}(|E|)$ since all nodes are visited from neighborhood to neighborhood to merge them into monochrome communities. Now considering the worst case for monochrome community reduced to a set of node singletons because the colors of all nodes are different, and assuming that at each step the new community merges only two communities, we deduce that the complexity is in $\mathcal{O}(|V|^2(|E| + |V| \log(|V|)))$.

5 Conclusion

We propose a new approach to detecting communities that relies on new criteria to identify them. Instead of a difference in connection density between the

community interior and its border, defining a community will minimize the chromatic entropy which is the entropy measure adapted to the problem of gathering nodes with the same colors.

This new paradigm provides an alternative approach to connectivity rule. It takes on its full meaning in challenges where the connection of nodes sharing the same property remains loose and therefore cannot be captured by an examination of the connection density as has been demonstrated for disease modules.

The significance of a community is assessed by its entropy. Low entropy means that the community structure cannot have been the result of chance confirming the intention behind its structure. We have proposed a CHROCODE heuristic solving this problem in polynomial time. The benchmarks detailed in [14] of this algorithm highlight the closeness of the two probability laws and very good performance of the algorithm.

A perspective would be to study how the grouping of nodes according to the major color rule could reveal connectivity patterns between nodes of the same color. Indeed, sharing the same property, these nodes could develop a particular connectivity structure characterizing a connection pattern that can be specific. Such a perspective would allow recognition of a property-dependent community through an hybrid model, combining the identical property recognition with connectivity rules for detecting meaningful communities.

References

1. Santo Fortunato and Darko Hric. Community detection in networks: a user guide. 659: 1–44. ISSN 0370–1573 (2016). <https://doi.org/10.1016/j.physrep.2016.09.002>, <https://www.sciencedirect.com/science/article/pii/S0370157316302964>
2. Liu, W., Pellegrini, M., Wang, X.: Detecting communities based on network topology. *Sci. Rep.* 4(1), 5739 (2014). ISSN 2045–2322. <https://doi.org/10.1038/srep05739>, <https://www.nature.com/articles/srep05739>
3. Fortunato, S.: Community detection in graphs. *Phys. Rep.* 486(3–5), 75–174 (2010). ISSN 03701573. <https://doi.org/10.1016/j.physrep.2009.11.002>, <http://arxiv.org/abs/0906.0612>
4. Khan, B.S., Niazi, M.A.: Network community detection: a review and visual survey. <http://arxiv.org/abs/1708.00977>
5. Newman, M.E.J.: Modularity and community structure in networks. *PNAS* 103(23): 8577–8582 (2006). <https://doi.org/10.1073/pnas.0601602103>, <https://www.pnas.org/doi/10.1073/pnas.0601602103>
6. Brandes, U., et al.: Maximizing Modularity is hard. <http://arxiv.org/abs/physics/0608255>
7. Pons, P., Latapy, M.: Computing communities in large networks using random walks (long version). <http://arxiv.org/abs/physics/0512106>
8. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. (10), P10008 (2008). ISSN 1742–5468. <https://doi.org/10.1088/1742-5468/2008/10/P10008>, <http://arxiv.org/abs/0803.0476>
9. Murata, T.: Detecting communities in social networks. In: Furtht, B., editor, *Handbook of Social Network Technologies and Applications*, pp. 269–280. Springer, US. ISBN 978-1-4419-7142-5. https://doi.org/10.1007/978-1-4419-7142-5_12

10. Ghiassian, S.D., Menche, J., Barabási, A.-L.: A DIseAse MOdule Detection (DIAMOnD) algorithm derived from a systematic analysis of connectivity patterns of disease proteins in the human interactome. **11**(4), e1004120 (2015). ISSN 1553–7358. <https://doi.org/10.1371/journal.pcbi.1004120>, <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004120>
11. Sharma, A., et al.: A disease module in the interactome explains disease heterogeneity, drug response and captures novel pathways and genes in asthma. **24**(11): 3005–3020. ISSN 0964–6906. <https://doi.org/10.1093/hmg/ddv001>, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4447811/>
12. Silberberg, Y., Kupiec, M., Sharan, R.: GLADIATOR: a global approach for elucidating disease modules. **9**(1), 48 (2017). ISSN 1756–994X. <https://doi.org/10.1186/s13073-017-0435-z>, <http://genomemedicine.biomedcentral.com/articles/10.1186/s13073-017-0435-z>
13. Pavan, S., Rommel, K., Marquina, M.E.M., Höhn, S., Lanneau, V., Rath, A.: Clinical Practice Guidelines for Rare Diseases: The Orphanet Database. **12**(1), e0170365 (2017). ISSN 1932–6203. <https://doi.org/10.1371/journal.pone.0170365>, <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0170365>
14. Delaplace, F.: Entropic detection of chromatic community structures (2023a). <https://hal.science/hal-04201260>, working paper or preprint
15. Chakrabarti, C.G., Chakrabarty, I.: Shannon entropy: axiomatic characterization and application. <http://arxiv.org/abs/quant-ph/0511171>
16. Shannon, C.E.: A mathematical theory of communication. **27**(3), 379–423 (1948). ISSN 0005–8580. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>
17. Delaplace, F.: Chrococ library in mathematica (2023). <https://doi.org/10.5281/zenodo.7767174>
18. Delaplace, F.: Chrococ module in Python (2023). <https://doi.org/10.5281/zenodo.7767111>



On the Hierarchical Component Structure of the World Air Transport Network

Issa Moussa Diop¹(✉), Cherif Diallo¹, Chantal Cherifi², and Hocine Cherifi³

¹ LACCA, Gaston Berger University, Saint-Louis, Senegal
diop.issa-moussa@ugb.edu.sn

² DISP LAB, University of Lyon 2, Lyon, France

³ LIB, University of Burgundy, Dijon, France

Abstract. In an era of global interconnectedness, transportation networks serve as vital conduits for international relations, trade, and cultural exchange. Previous research has extensively examined the worldwide air transportation network at three distinct levels: global, regional, and national, with contributions from various researchers shedding light on different facets of the network. Focusing on densely connected regions that extend from other dense areas, this study unveils the hierarchical structure of the global air transport network. Results emphasize the multi-level nature of the network, from significant regions to country-based divisions and ultimately to the intricate hub-and-spoke model. One observes increased hub dominance and negative assortativity values as one delves deeper into the network's hierarchy. This investigation brings a more comprehensive understanding of the global air transport network and its intricate hierarchical organization.

Keywords: Complex Networks · World Air Transportation Network · Hierarchical Component Structure

1 Introduction

In an era of global interconnectedness, transportation networks serve as the arteries of international relations, trade, and cultural exchange. The intricate web of airports worldwide facilitates the movement of people and goods across geographical boundaries. The air transportation network is subject to analysis at three levels: global, regional, and national. Various influential contributions have been made across these tiers, each shedding light on different aspects of the network.

The worldwide air transportation network has garnered substantial attention from researchers. [1] presents a comprehensive network examination, revealing power-law distributions in degree and betweenness centrality. This study contradicts the conventional notion that highly connected cities have correspondingly high betweenness centrality, suggesting geopolitical considerations play a significant role in network formation. [2] delve into the community structure, showcasing how communities align with geographical regions. They identify distinct

city roles within communities, shedding light on the complex network dynamics. [3] explore six node types, ranging from specific points like airports and cities to broader areas such as spatial regions around hubs, sub-national territories, and countries. Analyzing the networks, they find a common small-world and disassortative pattern across all node granularity levels. Additionally, they observe an increase in the clustering coefficient and a decrease in the average path length as node aggregation transitions from fine to coarse levels. Using the Louvain community detection method, the authors uncover consistently coherent community structures of approximately ten communities aligned with distinct geographical boundaries. [4] investigate the network's evolution, introducing the Global Airport Connectivity Index to gauge airport importance. This approach leads to classifying airports into regional and global hubs, demonstrating evolving strategies across different regions. [5] underscores the potential of component decomposition as a powerful tool for modeling and analyzing mesoscale network structures, extending beyond the world air transportation network. The authors investigate the unweighted, undirected world air transportation network. They find seven prominent spatially distinct local components with several smaller ones. Moreover, a global component covers all the world. A comprehensive comparative analysis of the component structure unveils the network's non-homogeneous topology, accentuating regional variations through local components and efficiency in inter-regional travel via the global component.

Regional analyses have been equally significant. [6] dissect the European airport network into core, periphery, and bridge layers, unraveling the network's core nodes' robustness. [7] study Southeast Asia's evolution, uncovering changing hub configurations over time.

Further attention extends to national networks. Italian, Indian, and Chinese networks have received extensive exploration. [8] reveal fractal structures within the Italian network. [9] studies India's network, unveiling hierarchical traffic corridors. Chinese network studies highlight its unique characteristics, driven by significant metropolises. US air transportation research showcases evolving trends toward hub-and-spoke structures.

Recent existing research shows that the mesoscopic dimension of networks is of prime interest to understand better their structure and dynamics[10–22]. It is particularly true for transportation networks where global, regional, and national characteristics are traditionally examined in isolation. Introducing the component structure paradigm reveals that airports across different regions interact due to economic, political, and historical factors. This study aims to delve further into these intricate connections within the global air transport network, focusing on densely connected regions. Through this approach, we seek a more comprehensive understanding of the air transport network's hierarchical structure through its dense areas and their interactions.

This paper proceeds as follows. Section 2 introduces the Data and analysis methodology. We then briefly describe the world transportation network's component structure and present the hierarchical structure uncovered from the components in Sect. 3. We discuss the results in Sect. 4 and conclude the paper in Sect. 5.

2 Data and Method

This section presents the dataset under test and the methodology to uncover the hierarchical component structure of the world air transportation network.

2.1 Data

Flight information from around the globe originates from FlightAware. The dataset spans six days between May 17, 2018, and May 22, 2018 [23]. Nodes correspond to an airport, and the links are direct flight connections between pairs of airports. The network contains 2734 airports interconnected by 16,665 direct flight routes. Table 1 reports its global topological characteristics.

Table 1. The topological features of the Worldwide Air Transportation Network. N is the network size. $|E|$ is the number of edges. μ is the density. d is the diameter. ζ is the transitivity. λ is the assortativity.

N	$ E $	μ	d	ζ	λ
2734	16665	0.004	12	0,26	-0,05

2.2 Methods

The Hierarchical component structure builds upon the concept of component structure. It consists in splitting a network into local and global components. The local components are the densely connected areas of the network. One can use a community or a multi-core-periphery structure analysis to uncover these dense parts. The remaining links with their nodes attached form the global components. This conceptual framework allows us to view a network as two sets of subnetworks. If we refer to the classical community structure representation the local components emerge from intra-community links, while the inter-community links give rise to the global components. Figure 1.A illustrates the three steps of revealing the component structure, including local and global components. An uncovered component is an isolated network. Consequently, one can extract its component structure revealing the first level of hierarchy, and so on. One iterates this process until reaching a stopping rule. Figure 1.B displays a network hierarchical structure illustrating this process. At level 0, there is the original graph. At level 1, there are three local components and one global component. The local component 3 doesn't include a component structure; it is a leaf. While local component 1 has three local components and one global component (level 2). Finally, local component 2 contains two local and one global component (level 2). Note that one can have several global components like in Fig. 1.A.

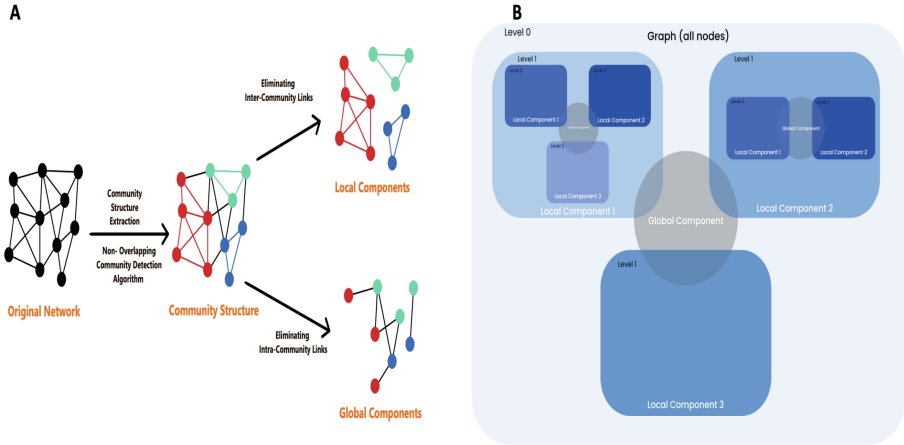


Fig. 1. A) Process to uncover the local components and the global components. B) Hierarchical component structure

One of the main advantages of the component structure representation is that it allows disentangling local interaction from global interactions. Furthermore, as each component is an isolated network, one can go deeper in its analysis, exploring it at various levels of granularity with the same tools in an iterative process. At this stage, one must consider a specific threshold to determine when to halt this iterative process. As we are interested in the network mesoscopic representation, the threshold parameter can be any particular measure, such as modularity, component size, or other relevant criteria.

In this study, the stopping criterion stipulates that a local component corresponds to a particular geographical area or its size should exceed 20 nodes. Until reaching this threshold condition, the analysis continues in-depth, resulting in the eventual formation of a hierarchical structure.

3 Experimental Results

3.1 Component Structure

Level 0 of the hierarchy consists of the component structure of the world air transportation network. It reveals twenty-seven local and nine global components. The local components include seven large components and twenty small components. Figure 2.A displays the large local components. They cover the following regions: North-Central America-Caribbean (657 airports), Europe (493 airports), East-Southeast Asia (416 airports), Africa-Middle East-Southern Asia (336 airports), Oceania (234 airports), South America (215 airports) and Russia-Central Asia-Transcaucasia (112 airports). The global components include one large component and eight small components. The large global components are distributed across the regions, as shown in Fig. 2.B. To extract the first level

of hierarchy, we focus on the components corresponding to geographical areas within the large components.

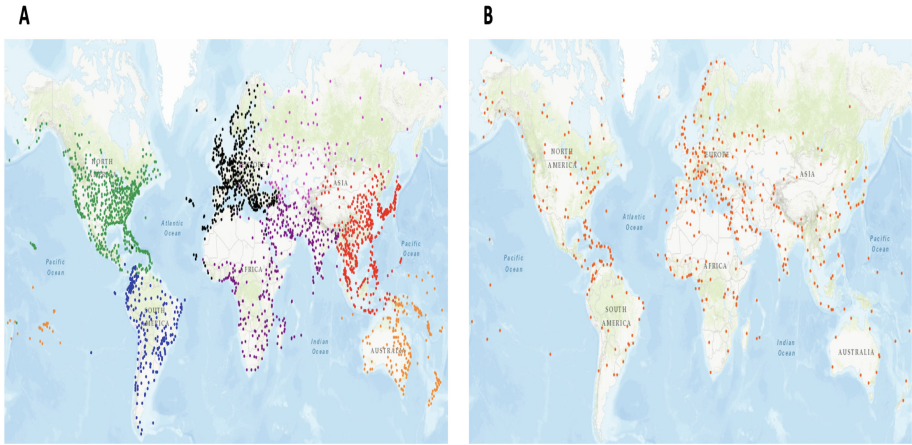


Fig. 2. A) represents the seven large local components. North-Central America-Caribbean is Green, Europe is black, East-Southeast Asia is red, Africa-Middle East-Southern Asia is magenta, Oceania is orange, South America is blue and Russia-Central Asia-Transcaucasia is pink. B) illustrates the large global components

3.2 First Level of Hierarchy

At level 1 of the hierarchy, one can distinguish two typical behaviors. Figure 3 shows a typical example of these two categories. In the first group, each component covers a well-delimited area, as illustrated in Fig. 3 A, showing the component structure of East and Southeast Asia. This category includes five local components (North-Central America-Caribbean, East-Southeast Asia, Africa-Middle East-Southern Asia, Oceania, and South America). The second group gathers the two other large local components (Europe and Russia-Central Asia-Transcaucasia). Figure 3.B illustrates the first level of hierarchy of the European component. One can notice that the components geographical localization is less pronounced.

In the first group, at level 1, we observe three classes. The first includes the North-Central America-Caribbean, East and Southeast Asia, and South America regions. They contain six components. The second Africa-Middle East-Southern Asia splits into seven components. Finally, Oceania has ten components. In the first class, countries with significant areas, such as the United States, China, and Brazil, are divided into components. Thus, the North-Central America-Caribbean components exhibit a modularity value of 0.38, revealing six distinct regions of airport connectivity. Among them, three areas are predominantly located within the United States, emphasizing the strong connections between

Alaska and the western parts of the United States due to their geographical proximity. Another cohesive component is formed by Mexico and Texas, driven by significant travel between Mexico and the United States through Texas. Notably, connections between Canada and Cuba are observed, possibly attributed to complex travel dynamics between the United States and Cuba, leading Cuban travelers to transit through Canada. A distinct Caribbean component emerges, encompassing some airports from the United States. China splits into East and Southeast Asia components, with a modularity of 0.38. One covers a significant part, and the other is in the West. Note that Japan and South Korea form a component. The Southeast Asian region is divided into three distinct components. The first one encompasses countries in the northern part of this region, including Thailand, Indonesia, Laos, Cambodia, Vietnam, and Myanmar. In addition, this part is very connected to Mongolia. It's worth noting that Mongolia's geographical proximity to China, coupled with various modes of transportation like trains, metros, vehicles, and air travel, makes air transport a more favorable option for connecting to other Asian countries. The second component includes Malaysia and Indonesia, while the Philippines stands apart as a separate entity due to its archipelago nature favoring air travel. With a modularity of 0.54, among the components in South America, two are located all along the east and West of Brazil. Venezuela-Colombia and Peru-Bolivia form distinct components. Chile and Argentina are well separated from other countries in this region.

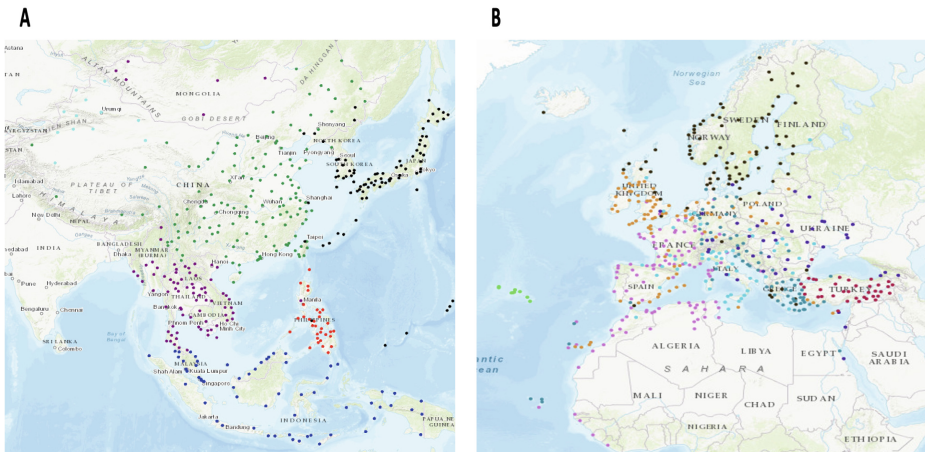


Fig. 3. The component from level 1. A) The components from East and Southeast Asia. All of them are well delimited. B) The components from the Europe. Only two are well delimited.

The West and East of Africa are very connected in the second class, with a modularity of 0.52. The South of Africa (Angola, Namibia, South Africa, etc.) is also a delimited component. Saudi Arabia, Egypt, Libya, and Sudan also

form a component. Except for Saudi Arabia and Iran, the major countries of the Middle East share the same component with the South of India. Indeed, Indians tend to travel to the United Arab Emirates to work. The other parts of India, some parts of Afghanistan, and the country of Bangladesh are in the same component. Iran and Madagascar each form separate components. In the last class, the modularity is 0.63 in the Oceania component. The components of this region are either countries or areas in Australia. Indeed, in Australia, four components are distributed between the West, North, Northeast, and South East Central. The other geographical components are countries such as New Zealand, French Polynesia, Vanuatu, Papua New Guinea, Solomon Islands, and an island group consisting of several countries.

In the second group, at level 1, one can see that in Fig. 3.B, only Turkey and the Scandinavian countries with some airports in the North of the UK and Greece are delimited areas in the European component. Indeed, the European Union is a single airspace, whereas Turkey is not part of the EU. In addition, the touristic areas in the islands in Portugal are separated from the rest. In Russia-Central Asia-Transcaucasia, the East of Russia and Kazakhstan are well separated from the other components.

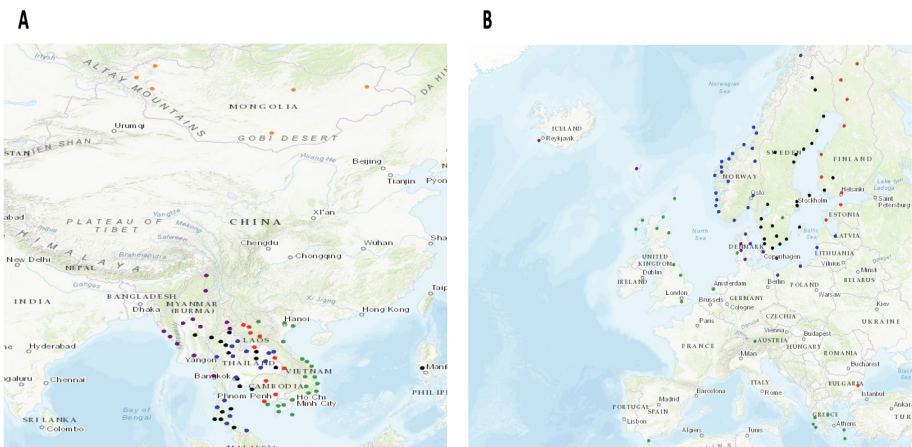


Fig. 4. The components from the Level 2. A) The components from Southeast Asian. B) The components from the Scandinavia countries.

The diameter in the first level of the hierarchy ranges between 3 and 6. In addition, the components corresponding to a country have a smaller diameter. Nevertheless, 5 or 6 jumps are necessary to reach two airports within their components in the North-Central America-Caribbean. Note that the Oceania region is very heterogeneous when you look at their diameters. The highest is 7, while the smallest is 2. Significantly, the components within Australia have large diameters. Entirely the components of the North-Central America-Caribbean are the

least dense. The South American components are the least transitive. Indeed, the components in Argentina and Colombia have almost no clusters. In the same way, the components located in the north of the United States and the West of China exhibit the same characteristics. The European components also contain numerous triangles. Some components in East and Southeast Asia are two times more transitive than others. Whereas for the other regions of level 1, the transitivity values of their components are a bit homogenous. Except for the European components, the components of this level are disassortative (with assortativity values ≤ -0.5), more than their parents. Like the assortativity, the hub dominance increases in the components of this level. Most of the highest hubs within their component can join more than half of the airports. Notably, Papua New Guinea and the Solomon Islands are centralized in the highest hubs (Port Moresby Jacksons Airport and Honiara Airport, respectively), which can join all the airports of these components. The global components of the various regions of this level have similar properties as the local components. Nevertheless, they are less disassortative and centralized than most local components. Note that in South America, there are more triangles in the global component than most of the local components.

3.3 Second Level of Hierarchy

At level 2, two North-Central America-Caribbean, East and Southeast Asia, and Africa-Middle East-Southern Asia components contain two subcomponents. Indeed, the Canada-Cuba, characterized by a modularity of 0.45, and Caribbean components with a modularity of 0.4 possess subcomponents corresponding to specific regions. The Canada-Cuba component is divided into five distinct subcomponents. Notably, Cuba's connection is focused on southeastern Canada, while other subcomponents exhibit more scattered connectivity. The Caribbean region reveals five components aligned with groups of countries. Countries like the Bahamas and Venezuela play dominant roles within their respective components. Figure 3.A shows that the Northern Southeast Asia components comprise the countries it encompasses. The modularity is 0.54. Components within this segment boast comparable sizes, with only Laos and Cambodia sharing a component. Additionally, Thailand is divided into two components, and Burma and Vietnam each form separate components. Moreover, Malaysia and Indonesia become separated with a modularity of 0.39. Indeed, Malaysia forms a component, while Indonesia has three delimited components in the East, Center, and North-West. Only two comprise subcomponents defining delimited regions among the Africa-Middle East-Southern Asia components. The Western and Eastern African component is subdivided into six regions, demonstrating a modularity of 0.46. The ECOWAS countries, excluding Nigeria, constitute one component, while Nigeria stands alone. Cameroon, Tanzania, and Kenya-Somalia-Djibouti follow the same pattern. Despite their distance, Ethiopia and Congo share a component, as Ethiopia often serves as a stopover en route to other parts of the world. With a modularity of 0.6, the Southern African component

is subdivided into six components, mainly corresponding to individual countries. Angola, South Africa, Mozambique, Botswana, and Namibia-Zimbabwe each constitute their component. Safaris have greatly enhanced these connections. The Peru-Bolivia component is separated into Peru and Bolivia at this level, with a modularity of 0.42. With a modularity of 0.44, Ecuador splits from Colombia. Colombia is divided into three components that don't correspond to geographical areas. There are four Oceania components which contain subcomponents. Indeed, the West of Australia consists of four components. While the Northeast of Australia also has five well-delimited components. These components are small, the largest of which includes ten nodes. With a modularity of 0.42, the Vanuatu region comprises three components that cover the North, Center, and South. The group of islands also splits into four components that regroup into a group of countries (islands). The most significant component among them contains seven airports. The Scandinavian component contains five components corresponding to countries or groups of countries. The modularity is 0.37. Indeed, Norway, Sweden, Finland, and Denmark become separate components as illustrated in Fig. 4.B. Especially the North of the United Kingdom and some airports in Greece share a component.

The diameter of the components from level 2 shows that to travel in a country, one needs 3 or 4 jumps at most. Exceptionally, 5 in the Bahamas and a component in Australia, or 2 in the components from the South of Africa and the Islands in Oceania. The components from level 2 located in the North America-Caribbean, East and Southeast Asia, and Africa-Middle East-Southern Asia tend to include more triangles. They are more transitive than their parent. Numerous components from level 2 located in Oceania don't contain any triangles. The components of level 2 tend to be more disassortative than their parent.

Moreover, the hub dominance increases for these components. For example, the following airports, such as Copenhagen Kastrup Airport in Denmark, Don Mueang Airport in Angola, and Jorge Chávez Airport in Peru, can join all the airports within their country. Don Mueang Airport can reach all airports in some parts of Thailand. By synthesizing these topological properties, several leaves of the hierarchical structure of the global air transport network have a star-based. They have transitivity close to zero or equal to zero, assortativity close to -1 or equal to -1, and hub dominance close to 1 or equal to 1. The global components also have these characteristics. But it's less prominent.

None of the level 2 components have sons. Indeed, their modularity is less than 0.3, or their size is less than 20 nodes. However, the hierarchical components of the world air transportation network have three levels of depth. In addition, none of the large global components have a hierarchical structure.

4 Discussion

This paper investigates the hierarchical structure of the world air transportation network through its component structure. We extract iteratively the local and global components from the components of the world air network corresponding to geographical regions. The component structure of the world air

transportation network exhibits seven significant regions and one global component. These regions correspond to North-Central America-Caribbean, Europe, East and Southeast Asia, Africa-Middle East-Southern Asia, Oceania, South America, and Russia-Central Asia-Transcaucasia. Thus, the analysis of airport networks across various global regions provides valuable insights into the intricate patterns of connectivity and the underlying factors shaping these networks.

Level 1 offers a closer examination of the components identified in Level 0. This level reveals regional patterns and divisions within the global air transportation network. We identify two groups. In the first group, we observe well-defined components. It includes North-Central America-Caribbean, East-Southeast Asia, Africa-Middle East-Southern Asia, Oceania, and South America. These components represented geographically cohesive regions, each with distinct connectivity characteristics. In North-Central America-Caribbean, the first level displays the high connectivity of various regions and parts of the United States. We observe non-obvious interconnections, such as between Canada and Cuba and some United States hubs with Caribbean airports. The East and Southeast Asia components regroup countries, even though China and Japan dominate their components. In addition, only the Philippines, in which islands favor air travel due to limited land connections, constitutes a single component. Mongolia, which borders China, is not connected to the latter. The Africa-Middle East-Southern Asia region reflects the intricate dynamics between West and East Africa, the Middle East and India, and the South of Africa. Only Iran and Madagascar form each one a component. Indeed, airports like Addis Ababa Bole Airport and Jomo Kenyatta Airport have essential hubs facilitating global connections. We observe a strong connection between Middle Eastern countries and South India, possibly due to employment-related travel. Oceania showcases a unique distribution of components within Australia and various island countries. Only one component includes several countries (Fiji, Marshall Islands, Micronesia, Kiribati, etc.). The South American region demonstrates how geographical characteristics and country relationships determine components. The clustering of Brazil's east and west, the distinctness of Chile and Argentina, and the presence of intermediary components like Venezuela-Colombia and Bolivia-Peru reveal the complex web of connections within the region. The second group includes Europe and Russia-Central Asia-Transcaucasia. Europe displays fewer well-delimited components, reflecting the unique characteristics of the European Union's unified airspace. Turkey and some Scandinavian countries form distinct components.

Level 2 focuses on subcomponents within the Level 1 components. It reveals finer-grained patterns of connectivity and division. For instance, within North-Central America-Caribbean, the Canada-Cuba and Caribbean components showcase subcomponents corresponding to specific regions. These subcomponents provide insights into the localized connectivity within these areas. In Southeast Asia, subcomponents highlight regional divisions, such as Laos and Cambodia, sharing a component, while Thailand is divided into two components. Africa-Middle East-Southern Asia has fewer subcomponents, but the Western

and Eastern African components exhibit distinct regions. Subdivision within the Southern African component reflects the impact of safaris on air travel, with separate components for individual countries. Oceania exhibits numerous sub-components, emphasizing its heterogeneity. Scandinavia splits into countries.

Diameter analysis reveals that regional flights are more efficient than inter-regional flights for the Africa-Middle East-Southern Asia and the South America regions. For these two areas, the diameter of the global component is equal to the diameter of the part. The opposite is true for the other sub-components, but not all. The components from the North America-Caribbean are less dense. The densest components from the different regions have a density of around 0.2. The sub-components in South America and mainly those in Australia are less transitive. In addition, finding a stopover in their global components is easier. Moving to Level 2, our analysis reveals even finer network connectivity details. Level 2 components in North America-Caribbean, East and Southeast Asia, and Africa-Middle East-Southern Asia tend to be more transitive, indicating increased clustering and interconnectedness compared to their parent Level 1 components. Conversely, several Level 2 components in Oceania lack triangles, suggesting reduced clustering. Hub dominance continues to increase at Level 2, with certain airports connecting all the airports within their respective countries. Several leaves of the hierarchical structure of the global air transport network exhibit star-based topological properties with low transitivity, high disassortativity, and dominant hubs. Global components exhibit similar properties to local components, albeit with reduced levels of disassortativity and centralization.

5 Conclusion

The investigation of the world air transport network based on its component structures, marks a pivotal step in understanding the network's complexity and interdependence. This paper highlights how geographical proximity, travel dynamics, and transportation infrastructure shape the structure of these components and how different regions exhibit diverse topological characteristics. This study uncovers three levels in the hierarchical structure of the world air transportation network. The first level covers large areas that are comparable to continents. The second level divides these large areas into groups of countries, except for Oceania and South America. Indeed, in Oceania, the presence of islands favors air transport. South American countries lack a strong integration policy. Moreover, the more we zoom in on the world's air transportation network, the more we see the hub-and-spoke character. Additionally, the negative assortativity and the hub dominance values increase. In future work, we plan to perform a comparative analysis with alternative hierarchical structure detection algorithms [24, 25].

References

1. Guimera, R., Amaral, L.A.N.: Modeling the world-wide airport network. *Eur. Phys. J. B* **38**(2), 381–385 (2004)
2. Guimera, R., Mossa, S., Turtschi, A., Amaral, L.A.N.: The worldwide air transportation network: anomalous centrality, community structure, and cities' global roles. In: *Proceedings of the National Academy of Sciences*, **102**(22), 7794–7799 (2005)
3. Sun, X., Wandelt, S., Zanin, M.: Worldwide air transportation networks: a matter of scale and fractality? *Transportmetrica A: Transp. Sci.* **13**(7), 607–630 (2017)
4. Cheung, T.K.Y., Wong, C.W.H., Zhang, A.: The evolution of aviation network: global airport connectivity index 2006–2016. *Transp. Res. Part E: Logist. Transp. Rev.* **133**, 101826 (2020)
5. Diop, I.M., Cherifi, C., Diallo, C., Cherifi, H.: Revealing the component structure of the world air transportation network. *Appl. Netw. Sci.* **6**(1), 1–50 (2021)
6. Lordan, O., Sallan, J.M.: Analyzing the multilevel structure of the European airport network. *Chinese J. Aeronaut.* **30**(2), 554–560 (2017)
7. Dai, L., Derudder, B., Liu, X.: The evolving structure of the southeast Asian air transport network through the lens of complex networks, 1979–2012. *J. Transp. Geogr.* **68**, 67–77 (2018)
8. Guida, M., Maria, F.: Topology of the Italian airport network: a scale-free small-world network with a fractal structure? *Chaos, Solitons Fractals* **31**(3), 527–536 (2007)
9. Bagler, G.: Analysis of the airport network of India as a complex weighted network. *Phys. A* **387**(12), 2972–2980 (2008)
10. Cherifi, H., Palla, G., Szymanski, B.K., Lu, X.: On community structure in complex networks: challenges and opportunities. *Appl. Netw. Sci.* **4**(1), 1–35 (2019)
11. Orman, K., Labatut, V., Cherifi, H.: An empirical study of the relation between community structure and transitivity. In: *Complex Networks*, pp. 99–110. Springer Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-30287-9_11
12. Gupta, N., Singh, A., Cherifi, H.: Community-based immunization strategies for epidemic control. In: *2015 7th International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–6. IEEE (2015)
13. Chakraborty, D., Singh, A., Cherifi, H.: Immunization strategies based on the overlapping nodes in networks with community structure. In: Nguyen, H.T.T., Snael, V. (eds.) *CSoNet 2016. LNCS*, vol. 9795, pp. 62–73. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42345-6_6
14. Kumar, M., Singh, A., Cherifi, H.: An efficient immunization strategy using overlapping nodes and its neighborhoods. In: *Companion Proceedings of the The Web Conference*, vol. 2018, pp. 1269–1275 (2018)
15. Lasfar, A., Mouline, S., Aboutajdine, D., Cherifi, H.: Content-based retrieval in fractal coded image databases. In: *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 1, pp. 1031–1034. IEEE (2000)
16. Demirkesen, C., Cherifi, H.: A comparison of multiclass SVM methods for real world natural scenes. In: Blanc-Talon, J., Bourennane, S., Philips, W., Popescu, D., Scheunders, P. (eds.) *ACIVS 2008. LNCS*, vol. 5259, pp. 752–763. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88458-3_68
17. Hamidi, M., Chetouani, A., El Haziti, M., El Hassouni, M., Cherifi, H.: Blind robust 3D mesh watermarking based on mesh saliency and wavelet transform for copyright protection. *Information* **10**(2), 67 (2019)

18. Orman, G.K., Labatut, V., Cherifi, H.: Towards realistic artificial benchmark for community detection algorithms evaluation. *Int. J. Web Based Commun.* **9**(3), 349–370 (2013)
19. Ghalmane, Z., Cherifi, C., Cherifi, H., El Hassouni, M.: Extracting backbones in weighted modular complex networks. *Sci. Rep.* **10**(1), 15539 (2020)
20. Rajeh, S., Savonnet, M., Leclercq, E., Cherifi, H.: Interplay between hierarchy and centrality in complex networks. *IEEE Access* **8**, 129717–129742 (2020)
21. Rajeh, S., Savonnet, M., Leclercq, E., Cherifi, H.: Characterizing the interactions between classical and community-aware centrality measures in complex networks. *Sci. Rep.* **11**(1), 10088 (2021)
22. Rajeh, S., Savonnet, M., Leclercq, E., Cherifi, H.: Comparative evaluation of community-aware centrality measures. *Quality Quantity* **57**(2), 1273–1302 (2023)
23. Alves, L.G.A., Aleta, A., Rodrigues, F.A., Moreno, Y., Amaral, L.A.N.: Centrality anomalies in complex networks as a result of model over-simplification. *New J. Phys.* **22**(1), 013043 (2020)
24. Schaub, M.T., Li, J., Peel, L.: Hierarchical community structure in networks. *Phys. Rev. E* **107**(5), 054305 (2023)
25. Polanco, A., Newman, M.E.J.: Hierarchical core-periphery structure in networks. arXiv preprint [arXiv:2301.03630](https://arxiv.org/abs/2301.03630) (2023)



Weighted and Unweighted Air Transportation Component Structure: Consistency and Differences

Issa Moussa Diop¹✉, Cherif Diallo¹, Chantal Cherifi², and Hocine Cherifi³

¹ LACCA, Gaston Berger University, Saint-Louis, Senegal
diop.issa-moussa@ugb.edu.sn

² DISP LAB, University of Lyon 2, Lyon, France

³ LIB, University of Burgundy, Dijon, France

Abstract. The topological structure of the world air transportation network is the subject of much research. This paper reports a comparative analysis of the weighted and unweighted air transportation network at the mesoscopic level. We use the component structure to isolate regional from interregional traffic and infrastructure. Recently introduced in the network literature, the component structure splits the network into local and global components. The local components are the dense parts of the network. They capture the regional flights. The global components linking the dense parts capture the inter-regional flights. Results display fewer local components well delimited and more global components covering the world than the unweighted world air transportation network. Beyond their structural implications, these components offer practical advantages. They can be a foundation for optimizing transportation routes and schedules, leading to cost savings and reduced travel times. Stakeholders in transportation, including airlines, shipping companies, urban planners, and policymakers, can leverage this knowledge to make informed decisions and strategic plans that promote economic growth and environmental sustainability.

Keywords: Complex Networks · World Air Transportation Network · Weighted network · Unweighted network

1 Introduction

The air transport system connects all countries in the world. This infrastructure has a direct impact on society and the global economy. Indeed, millions of people and goods transit through the air every day. Recently, we have seen how the COVID-19 pandemic spread rapidly because of air transport. Thus, understanding the air transportation system can help policymakers make decisions that can improve or affect this system. Network science provides a simple way to represent and understand numerous networked structures, such as infrastructure and social networks, and more particularly their mesoscopic characteristics [1–12]. Thus, several studies are devoted to the air transportation network, including structure, dynamics, and robustness.

The literature reports numerous studies of unweighted air transportation networks [13–16]. They cover national, regional, and worldwide networks and include investigations at the macroscopic, mesoscopic, and microscopic scales. All these networks share some common characteristics. They are generally small-world and scale-free. Moreover, some exhibit a community structure [13].

Most of the works in the weighted air transportation network focus on weighted national air transportation networks. Indeed, [16] study the weighted airport network of India. In [17], the authors investigate the evolution of the air transportation network of the United States from 2002 to 2005. The authors in [18] study the Australian airport network's structure and dynamic flow. In [15], the authors analyze the characteristics of the Asian international passenger aviation market in 2014 and 2018. In these studies, several consistent findings emerge. Firstly, the airports in various networks exhibit a scale-free structure, with a few highly connected hubs dominating the network. Secondly, a strong correlation exists between node strength (measuring importance) and degree (number of connections), indicating that highly connected airports also handle a significant traffic volume. Thirdly, the rich-club phenomenon highlights substantial traffic flow among interconnected hubs, improving the network's efficiency. Additionally, the incorporation of weighted edges underscores the role of low-weighted connections in forming topological clusters. Finally, the studies emphasize the dynamic nature of air transportation networks, influenced by factors like the growth of Low-Cost Carriers and the impact of developing countries on regional air transport networks.

In a previous study, we analyzed the world air transportation network through a new mesoscopic structure called the component structure [19]. A network contains two types of components. The dense parts of the network form the local components. The interactions between the local components are called the global components. Therefore, one must extract the dense areas to build the component structure. The community structure or the core-periphery algorithms are good candidates to do so. Indeed, the communities constitute cohesive groups of nodes sparsely connected [20–22]. The core-periphery [23–25] structure contains two groups of nodes (core and periphery). The core nodes are tightly connected. The periphery nodes are almost not connected. The links between the core and periphery nodes are relatively dense. Networks can exhibit a multi-core-periphery structure [26]. One can extract the dense parts using any algorithm to uncover the communities or the various cores to form the local components. In the previous study, we analyzed the unweighted world transportation network. Results show that the local components capture the regional destinations, while the global components represent the inter-regional flights. Our investigations use no information on the dynamic in the infrastructure. This simplification can lead to centrality anomalies [11, 26] and hide critical information about the flow of flights and passengers in the infrastructure. Indeed, the traffic in the various routes can be pretty different, and failing to integrate these differences can lead to misleading conclusions.

In this paper, we perform an extensive comparative analysis of the route (unweighted) and traffic (weighted) networks at the mesoscopic level to better understand their differences and consistencies. The network of 2734 nodes and 16665 links originates from FlightAware [27]. It collects the flights between May 17, 2018, and May 22, 2018 [28]. Nodes represent airports, and links are direct flights between air-

ports. The link weight is the number of flights between two airports. Table 1 reports basic properties (unweighted and weighted). As the component structure has shown promise in disentangling regional with inter-regional routes [19], we adopt this representation. We rely on community detection algorithms to extract the component structure. It allows us to explore the traffic on the different regional and inter-regional infrastructures. Furthermore, we compare the weighted and unweighted topological properties at the macroscopic level.

The rest of the paper is organized as follows. Section 2 examines the network mesoscopic structure. Starting with the community structure, we perform an extensive comparative evaluation of the various weighted and unweighted local and global components. Section 3 explores the topological properties of the local and global components. Finally, we conclude in Sect. 4.

Table 1. Basic topological properties of the world air transportation network. N is the network size. $|E|$ is the number of edges. d is the diameter. L is the average shortest path length. μ is the density. ζ and ζ^w are, respectively, the unweighted and weighted average clustering coefficients. λ and λ^w are respectively the unweighted and weighted assortativity, also called the degree-degree correlation coefficient. η is the hub dominance.

	N	$ E $	d	L	μ	ζ	ζ^w	λ	λ^w	η
network	2734	16665	12	3,86	0,004	0,046	0,007	-0,046	0,048	0.09

2 Mesoscopic Structure Analysis

Our main goal is to explore the regional and interregional differences and similarities of the world air transportation network. So, the first step is to split it into various regions. Rather than relying on the classical IATA subdivision based on geographic considerations, we consider the different mesoscopic representations of the weighted and unweighted world transportation networks. We extract the dense part of the networks using a community detection algorithm. Consequently, first, we investigate the weighted and unweighted community structures. We also explore their variations linked to the community detection algorithm. Then, we turn to the component structure to perform an extensive comparative analysis.

2.1 Community Structure

Influence of the Community Detection Algorithm. To evaluate the influence of the community detection algorithm, we also perform the community detection of the weighted network using the Combo algorithm. The number of communities extracted from the networks by the algorithms is quite different. Louvain uncovers 17 communities with sizes ranging from 725 to two airports. In contrast, Combo identifies only seven. The largest community includes 703 airports, and the smallest consists of 70 airports. Table 2 reports the quality metrics of the two community structures. Their modularity is identical (0.47). Its value indicates that the communities are dense, with a

medium proportion of inter-community links. The community structure of Louvain contains few connections between the communities with large weights. In contrast, Combo reveals more inter-community links with smaller weights. In both cases, the mixing parameter values demonstrate that the communities are well-separated.

Table 2. Quality metrics of the community structures uncovered by Louvain and Combo community detection algorithms: Modularity, Mixing parameter, NMI

	Modularity	Mixing parameter	NMI
Louvain	0,47	0,043	0,87
Combo	0,47	0,046	

Although Louvain uncovers more communities than Combo, their community structures have numerous similitudes. Indeed, Combo merges some communities of Louvain. A high value of the NMI (0.87) confirms their similarity. Five communities are very similar. They cover the same geographical areas. In Fig. 1, these areas correspond to the communities with the same color. They are in North and Central America-Caribbean, Europe-Russia-Central Asia-Central Asia, East and Southeast Asia-Oceania, Africa-Middle East-Southern Asia, and South America. Table 3 reports their Jaccard Index. Its value for all similar communities is higher than 0.85. For North and Central America-Caribbean and Europe-Russia-Central Asia-Central Asia, it is higher than 0.9, indicating a high similarity between the communities uncovered by the algorithms. The thirteen other communities extracted by Louvain, mainly located in Canada and Alaska, are merged by Combo. To sum up, although the community structures uncovered by Louvain and Combo differ, the large communities have much in common. We adopt the Louvain communities to build the component structure used in further investigations.

Table 3. The Jaccard Index of the five similar communities uncovered by Louvain and Combo.

Community	Jaccard Index
North and Central America-Caribbean	0,94
Europe-Russia-Central Asia-Central Asia	0,98
East and Southeast Asia-Oceania	0,85
Africa-Middle East-Southern Asia	0,88
South America	0,85

2.2 Component Structure Analysis

We distinguish two types of components. Large components include more than 100 airports, while small components' size is below this threshold. In this section, we describe their features in the weighted network. Furthermore, we compare them with their analog uncovered in the unweighted network [19].

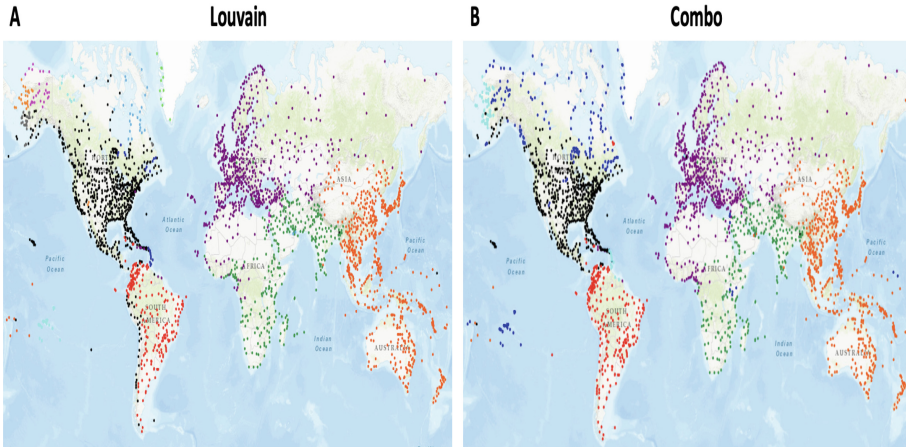


Fig. 1. A) Illustrates the communities identified through the Louvain community detection algorithm, encompassing eighteen communities. B) Showcases the communities revealed by the Combo community detection algorithm, comprising seven communities. Each color corresponds to a specific community in both cases, and similar communities are represented by matching colors.

Local Components. The local components correspond to the 17 communities uncovered by Louvain in the weighted network. There are five large and twelve small local components.

The large local components cover 1) North and Central America-Caribbean (725 airports), 2) Europe-Russia-Central Asia-Central Asia (683 airports), 3) East-Southeast Asia-Oceania (630 airports), 4) Africa-Middle East-Southern Asia (313 airports), and 5) South America (201 airports). Altogether, they regroup more than 93% of the world's airports. Like in the unweighted local components, the large weighted local components do not reflect strict geographical divisions. They correspond more to political, cultural, historical, and economic divides. For example, all the airports in Morocco, Tunisia, Benin, and Cameroon belong to the European component. It is because of the solid economic and historical ties these countries share with Europe.

One can distinguish two typical behaviors when comparing the large local components of the weighted and unweighted network illustrated in Fig. 2. In the first case, the components are very similar. In the second case, separated components in the unweighted network merge into a single component.

One can distinguish two typical behaviors when comparing the large local components of the weighted and unweighted network illustrated in Fig. 2. In the first case, the components are very similar. In the second case, separated components in the unweighted network merge into a single component.

There are three similar components (North and Central America-Caribbean, Africa-Middle East-Southern Asia, and South America). We quantify their similarity using the Jaccard index. The higher the similarity, the closer the Jaccard index is to 1.

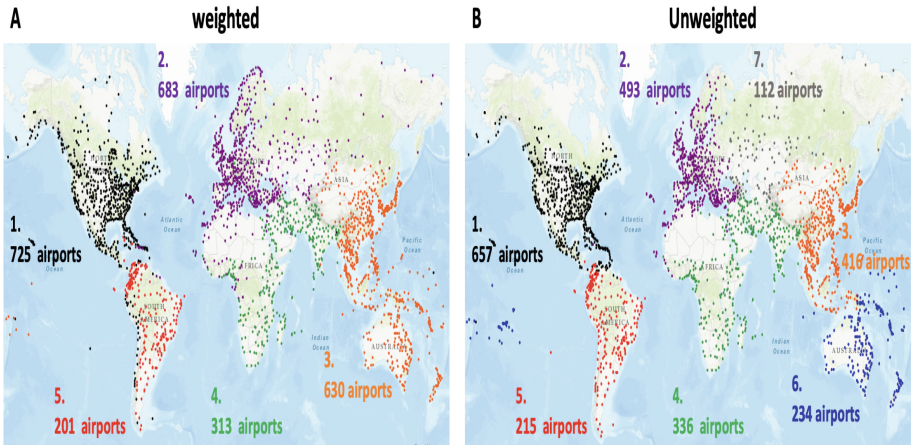


Fig. 2. A) depicts the airports within the large components of the weighted network. B) shows the airports within the large components of the unweighted network. Each color corresponds to a distinct component. The North and Central America-Caribbean component is represented by black (1). The Africa-Middle East-India component is denoted by green (4). The South American component is highlighted in red (5). The magenta represents the Europe-Russia-Central Asia-Central Asia component (4). In the unweighted network, it splits into magenta (4) in Europe and gray (7) in Russia. The East and Southeast Asia-Oceania component in orange (5) and, in the unweighted network, divides into orange (5) in East and Southeast Asia and blue (6) in Oceania.

The weighted component encompassing North and Central America-Caribbean boasts 10% more airports compared to its unweighted counterpart. Notably, their Jaccard index is relatively high at 0.81, highlighting significant commonality regarding shared airports. However, it's essential to recognize that 39 airports in the unweighted component do not appear in the weighted one. These omissions primarily affect airports in the French Antilles and Venezuela. Strikingly, even John F. Kennedy Airport is absent from the weighted component.

Conversely, the weighted component introduces a set of new airports totaling 107. These additions are predominantly located in Canada, Alaska, Peru, and Chile. Notably, London Heathrow Airport, a pivotal air hub in the United Kingdom, appears in the weighted component. This inclusion is justified by the airport's extensive connections to the United States, with numerous flights from North America arriving at Heathrow. The most unexpected inclusions in this weighted component are the Marshall Airport in the Marshall Islands and the Osmany Airport in Bangladesh. The Marshall Airport collaborates with United Airlines, a major U.S. carrier, strengthening its connectivity with the United States. On the other hand, Osmany Airport facilitates flights to and from London Heathrow.

The Jaccard index between the unweighted (336 airports) and weighted (313 airports) Africa-Middle East-Southern Asia component is high, registering at 0.87. 35 airports in the unweighted component are absent in the weighted one, primarily in the Middle East and West Africa. Conversely, eleven new airports have emerged in

the weighted component, with 6 located in Kenya, 2 in West Africa, and 3 in Europe. Unexpectedly, Germany's largest airport, Frankfurt am Main, is part of this component, owing to its substantial air traffic connections with Saudi Arabia and India. Poland's Rzeszów-Jasionka Airport and Greece's Araxos Airport also feature in this component due to their robust traffic ties with Frankfurt am Main.

The Jaccard Index between the unweighted (215 airports) and weighted (201 airports) South America components is 0.78. Thirty-two airports from Chile and Peru vanished in the weighted component, shifting to the weighted North and Central America-Caribbean region. Meanwhile, the weighted South America component now includes Venezuela, Colombia, and Cuba airports.

Two merged components are formed: "Europe-Russia-Central Asia", combining the unweighted European and Russia-Central Asia-Transcaucasia components, and "East and Southeast Asia-Oceania", comprising the unweighted East and Southeast Asia and Oceania components. This allows us to compare them with their weighted counterparts.

The Jaccard Index for the European-Russia component is 0.87. Nine airports from the unweighted version vanish, while 87 new ones emerge in the weighted component, mainly in Norway, West and North Africa, Iran, the United Arab Emirates, and the French Antilles. Beyond geography, these countries share political, historical, and economic ties. Moscow and St Petersburg airports connect Russia with Europe and other world regions. North Africa, West Africa, and the French Antilles have strong political and historical links with Europe, resulting in substantial traffic between these regions.

The unweighted and weighted East and Southeast Asia-Oceania components exhibit a high similarity with a Jaccard index of 0.97. These regions, connected by major airports, have significant traffic. In the weighted component, twenty-three airports in French Polynesia disappeared, while three airports in Russia and India emerged. Vladivostok and Yuzhno-Sakhalinsk airports facilitate exchanges with China, Japan, and South Korea. Trichy airport in India has heavy traffic with Malaysia, Singapore, and Sri Lanka.

Figure 3.A displays small local components in both the weighted (A) and unweighted (B) networks for comparison. These components are either within a single country or span a few countries or subregions. In the weighted network, these small components range from 30 to 2 airports. Five similar components exist in both the weighted and unweighted networks. Additionally, some small components appear or disappear in the weighted network compared to the unweighted one. Notably, Alaska and Greenland have isolated areas with limited connections and flights, making air transport crucial. French Polynesia and French Antilles-Ontario-Quebec regions have two small components each, connecting airports globally despite sparse connections. These areas heavily rely on air travel due to their island nature and community ties. Thirteen small unweighted components integrate into the weighted large components, with four in North America, five in Europe, and four in Africa.

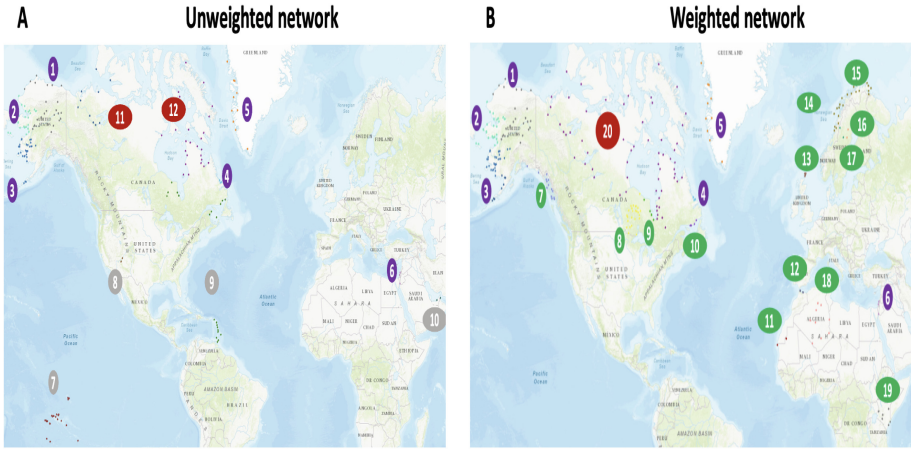


Fig. 3. Small local components are derived from the weighted (A) and unweighted (B) world air transportation networks. The magenta components are consistent in both networks. Components colored in beige are exclusive to the unweighted network, while green components are exclusive to the weighted network. The red component (20) is divided into two (11 and 12) in the weighted network. Note that geographical areas outside the figure do not include small components.

Globally, small local components represent 6.6% of airports. They are in North and Central America-Caribbean (3 in Alaska, 2 in Canada, and 1 in the Caribbean), Europe (1 in Greenland and 1 in Israel), East and Southeast Asia-Oceania (1 in French Polynesia and 1 in Australia), and Africa-Middle East-Southern Asia (1 in the United Arab Emirates). Five of these components have fewer than five airports.

Global Components. Figure 4 shows the global components extracted from the weighted world air transportation network. There are one large and 11 small global components. The small global components include 36 airports in North and Central America-Caribbean and East and Southeast Asia-Oceania. Their size ranges from 4 airports to 2 airports. Canada contains most of them (11). Only two components are shared with the unweighted world air transportation network. In the following, we neglect these small components. The global component encompasses 557 airports (20.44% of world airports) and spans worldwide.

The Jaccard Index between the weighted and unweighted large global components is low at 0.63. The weighted large global component has 14 more airports than the unweighted one, but they differ significantly. Specifically, 100 airports disappear in the weighted component, while 144 new ones join. This shift occurs because Oceania and Russia merge with neighboring regions in the weighted network, reducing inter-component connections. New airports come into play due to unique features in North and Central America-Caribbean, Europe-Russia-Central Asia, and Africa-Middle East-India components. Highly connected airports like London Heathrow and John F. Kennedy, not originally in their geographical components, enhance local component

links, drawing new airports into the global component. Such airport transfers profoundly alter the connected component.

3 Global Topological Properties of the Components

3.1 Clustering Coefficient

One can define $C(k)$ and $C^w(k)$ as the average clustering coefficient of nodes with degree k for, respectively, unweighted and weighted networks. The relation $C^w > C$ indicates that the interconnected triplets tend to be formed by links with high weights. The opposite $C^w < C$ shows that lower-weight edges produce interconnected triplets.

Let's delve into the large local components. The average unweighted clustering coefficient inversely relates to node degree. Lower-degree nodes exhibit higher clustering coefficients, while hubs are less cohesive. When fitting $C(k)$ with the law $k^{-\gamma}$, we find $\gamma \approx 0.3$ for Africa-Middle East-Southern Asia and South America components, and around $\gamma \approx 0.2$ for others.

In contrast, the average weighted clustering coefficient remains constant regardless of degree. Furthermore, the weighted average clustering coefficients are consistently lower than their unweighted counterparts. This indicates that interconnected triplets in large local components tend to form through low-weight edges.

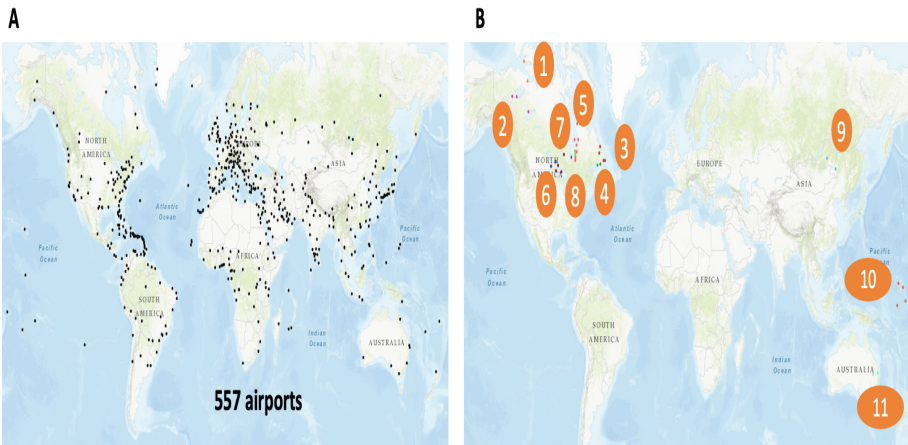


Fig. 4. A) The airports in the large global component. B) The 11 small global components are circled. The size range are between 2 and 3 airports.

In general, the large global component behaves similarly to local components. In the unweighted network, the fitted power law function has a $\gamma \approx 0.58$ coefficient. Consequently, the global component has fewer interconnected triplets than the local ones. This is because the global component includes long-distance traffic airports, many of which

are hubs in their country with limited rerouting capacity in a hub-and-spoke configuration. Like in large local components, the unweighted average clustering coefficient for a node with degree k is higher than the weighted average clustering coefficient. This indicates that connected triplets primarily involve nodes with low-weight edges.

3.2 Strength Distribution

We perform a goodness of fit evaluation of the strength distributions of the large local components with the Kolmogorov Smirnov test (KS). Results reported in Table 4 reveal that the log-normal distribution better fits the large local components. Moreover, they are heavy-tailed, like the degree distribution of the unweighted large local components of the world air transportation network. One can expect this result. Indeed, the higher the node degree, the higher its weight.

The relationship between strength and degree in each large local component is obvious, described by the equation $s(k) = k^\beta$. Despite β varying slightly between 2.1 and 2.3 for all large local components, this pattern aligns with findings from [16], where $\beta \approx 1.43$ was reported.

Table 4. The Kolmogorov-Smirnov goodness-of-fit values (KS-test) are computed for the degree distribution, considering distributions such as Power-law, Truncated power law, Log-Normal, and Stretched exponential. The smallest value, indicated in bold, signifies the best fit

	Power law	Truncated Power law	Log-Normal	Stretched Exponential
North and Central America-Caribbean	0.31	0.136	0.044	0.06
Europe-Russia-Central Asia	0.31	0.27	0.041	0.06
East and Southeast Asia	0.32	0.15	0.032	0.05
Africa-Middle East-Southern Asia	0.32	0.15	0.045	0.065
South America	0.33	0.15	0.043	0.18

Three categories of large local components based on the strength-degree curve exponent (β) can be distinguished. The Europe-Russia-Central Asia component, the first category, features multiple hubs with nearly equal traffic. Traffic increases gradually with degree. The North-Central America-Caribbean and East-Southeast Asia-Oceania components fall into the second category. These components also host numerous hubs, but traffic is highly concentrated in a few of them. The last category includes the Africa-Middle East-Southern Asia and South America components. These have fewer hubs and less traffic compared to the other types. The strength increases rapidly with degree, with only a few hubs accumulating the most traffic.

The strength distribution of the Large Global Component also follows a log-normal law with a heavy-tailed pattern akin to the large local components. Similar distribution parameters are observed in the East-Southeast Asia-Oceania component. Higher node degrees correlate with more traffic accumulation, with airports having at least 100 flights, which differs from the large local components. The β exponent aligns with that of North and Central America-Caribbean and the East-Southeast Asia-Oceania components.

4 Discussion and Conclusion

This paper investigates the relationship between the weighted and unweighted world-wide air transport network and its impact on its component structure.

In summary, the weighted network has fewer components than the unweighted one. Large local components in both cases cover distinct geographic areas, with some variation. The weighted network has five large local components, while the unweighted network has seven. Three components are similar (North and Central America-Caribbean, Africa-Middle East-South Asia, South America), while the other two (Europe-Russia-Central Asia, East-South-East Asia-Oceania) combine neighboring components from the unweighted network due to substantial traffic and economic integration. Major airports also integrate regions with shared flight traffic, like John F. Kennedy Airport in Europe-Russia-Central Asia, London Heathrow in North and Central America-Caribbean, and Frankfurt airports in Africa-Middle East-South Asia.

Although the weighted network has more global components, both networks have a single large global component, each comprising about 20% of the world's airports. However, their content differs significantly. Merging local components leads to the disappearance of many connecting airports in the global component, while new airports emerge when an airport is not in its natural geographic area. For example, most airports linked to John F. Kennedy outside the Europe-Russia-Central Asia component integrate the global component.

The analysis of average weighted clustering coefficients in large local components shows they remain constant regardless of node degree, consistently lower than their unweighted counterparts. Less-trafficked airports form triplets within these components. Conversely, unweighted clustering coefficients decrease as node degree rises, indicating higher-degree nodes have lower clustering coefficients, highlighting issues with Hub and Spoke configurations for rerouting. Africa-Middle East-Southern Asia and South America components have fewer interconnected hubs, resulting in a less pronounced hub-and-spoke setup. The global component displays a similar pattern but with a stronger hub-and-spoke effect due to more hubs and fewer triplets, intensifying rerouting inefficiencies.

Strength distributions follow a Log-Normal pattern with heavy tails. Hubs manage substantial traffic in Europe-Russia-Central Asia, North-Central America-Caribbean, and East-Southeast Asia-Oceania. In contrast, fewer hubs in Africa-Middle East-Southern Asia and South America handle significant flight volumes. The global component exhibit similar Log-Normal strength distributions, with hubs managing most traffic.

This analysis demonstrates how representing the world air transportation network through its component structure is crucial for uncovering regional and inter-regional similarities and differences. Traditional network metrics are designed for homogeneously dense networks, making it vital to separate local and global analysis due to local density variations. This approach paves the way for various research opportunities.

References

1. Orman, K., Labatut, V., Cherifi, H.: An empirical study of the relation between community structure and transitivity. In: *Complex Networks*, pp. 99–110. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-30287-9_11
2. Gupta, N., Singh, A., Cherifi, H.: Community-based immunization strategies for epidemic control. In: *2015 7th International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–6. IEEE (2015)
3. Chakraborty, D., Singh, A., Cherifi, H.: Immunization strategies based on the overlapping nodes in networks with community structure. In: Nguyen, H.T.T., Snasel, V. (eds.) *CSoNet 2016*. LNCS, vol. 9795, pp. 62–73. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42345-6_6
4. Kumar, M., Singh, A., Cherifi, H.: An efficient immunization strategy using overlapping nodes and its neighborhoods. In: *Companion Proceedings of the Web Conference*, vol. 2018, pp. 1269–1275 (2018)
5. Lasfar, A., Mouline, S., Aboutajdine, D., Cherifi, H.: Content-based retrieval in fractal coded image databases. In: *Proceedings 15th International Conference on Pattern Recognition, ICPR-2000*, vol. 1, pp. 1031–1034. IEEE (2000)
6. Demirkesen, C., Cherifi, H.: A comparison of multiclass SVM methods for real world natural scenes. In: Blanc-Talon, J., Bourenane, S., Philips, W., Popescu, D., Scheunders, P. (eds.) *ACIVS 2008*. LNCS, vol. 5259, pp. 752–763. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88458-3_68
7. Hamidi, M., Chetouani, A., El Haziti, M., El Hassouni, M., Cherifi, H.: Blind robust 3D mesh watermarking based on mesh saliency and wavelet transform for copyright protection. *Information* **10**(2), 67 (2019)
8. Orman, G.K., Labatut, V., Cherifi, H.: Towards realistic artificial benchmark for community detection algorithms evaluation. *Int. J. Web Based Commun.* **9**(3), 349–370 (2013)
9. Ghalmane, Z., Cherifi, C., Cherifi, H., El Hassouni, M.: Extracting backbones in weighted modular complex networks. *Sci. Rep.* **10**(1), 15539 (2020)
10. Rajeh, S., Savonnet, M., Leclercq, E., Cherifi, H.: Interplay between hierarchy and centrality in complex networks. *IEEE Access* **8**, 129717–129742 (2020)
11. Rajeh, S., Savonnet, M., Leclercq, E., Cherifi, H.: Characterizing the interactions between classical and community-aware centrality measures in complex networks. *Sci. Rep.* **11**(1), 1–15 (2021)
12. Rajeh, S., Savonnet, M., Leclercq, E., Cherifi, H.: Comparative evaluation of community-aware centrality measures. *Quality Quantity* **57**(2), 1273–1302 (2023)
13. Guimera, R., Mossa, S., Turtschi, A., Amaral, L.A.N.: The worldwide air transportation network: anomalous centrality, community structure, and cities' global roles. *Proc. Natl. Acad. Sci.* **102**(22), 7794–7799 (2005)
14. Zanin, M., Lillo, F.: Modelling the air transport with complex networks: a short review. *Eur. Phys. J. Spec. Top.* **215**(1), 5–21 (2013)
15. Cheung, T.K.Y., Wong, C.W.H., Zhang, A.: The evolution of aviation network: global airport connectivity index 2006–2016. *Transp. Res. Part E: Logist. Transp. Rev.* **133**, 101826 (2020)
16. Bagler, G.: Analysis of the airport network of India as a complex weighted network. *Phys. A* **387**(12), 2972–2980 (2008)
17. Xu, Z., Harriss, R.: Exploring the structure of the U.S. intercity passenger air transportation network: a weighted complex network approach. *GeoJournal* **73**(2), 87 (2008)
18. Md Murad Hossain and Sameer Alam: A complex network approach towards modeling and analysis of the Australian airport network. *J. Air Transp. Manag.* **60**, 1–9 (2017)

19. Diop, I.M., Cherifi, C., Diallo, C., Cherifi, H.: Revealing the component structure of the world air transportation network. *Appl. Netw. Sci.* **6**(1), 1–50 (2021)
20. Fortunato, S., Hric, D.: Community detection in networks: a user guide. *Phys. Rep.* **659**, 1–44 (2016)
21. Cherifi, H., Palla, G., Szymanski, B.K., Lu, X.: On community structure in complex networks: challenges and opportunities. *Appl. Netw. Sci.*, **4**(1), 1–35 (2019)
22. Newman, M.E.J.: Detecting community structure in networks. *Eur. Phys. J. B* **38**(2), 321–330 (2004)
23. Borgatti, S.P., Everett, M.G.: Models of core/periphery structures. *Soc. Netw.* **21**(4), 375–395 (2000)
24. Lee, S.H., Cucuringu, M., Porter, M.A.: Density-based and transport-based core-periphery structures in networks. *Phys. Rev. E* **89**(3), 032810 (2014)
25. Zhang, X., Martin, T., Newman, M.E.J.: Identification of core-periphery structure in networks. *Phys. Rev. E* **91**(3), 032803 (2015)
26. Kojaku, S., Masuda, N.: Finding multiple core-periphery pairs in networks. *Phys. Rev. E* **96**(5), 052313 (2017)
27. Flightaware. <https://flightaware.com/>
28. Alves, L.G.A., Aleta, A., Rodrigues, F.A., Moreno, Y., Amaral, L.A.N.: Centrality anomalies in complex networks as a result of model over-simplification. *New J. Phys.* **22**(1), 013043 (2020)



Effects of Null Model Choice on Modularity Maximization

Christopher Brissette, Ujwal Pandey, and George M. Slota^(✉)

Rensselaer Polytechnic Institute, Troy, NY, USA
{brissc,pandeu,slotag}@rpi.edu

Abstract. Given a defined set of communities, modularity is computed by comparing each existing edge with its probability of occurrence in a random graph null model. The heuristic has historically garnered a wealth of attention, and many community detection algorithms have been designed around maximizing modularity. Despite this, there are potential issues with the Chung-Lu null graph model that underpins the heuristic. In this manuscript, we explore the output communities given by modularity maximization when this null model is subject to change. We construct two null models using iterated double edge swapping and maximum likelihood estimation, and we use these models as the basis for new modularity-like heuristics we call *desmod*, and *mlemod*. We compare the clusters output by standard modularity maximization with those output by our methods on a test suite of LFR benchmark graphs and find that changing the null model consistently increases the normalized mutual information scores when the mixing parameter is high.

Keywords: clustering · modularity · random graphs

1 Introduction

Community detection, also synonymously referred to as ‘graph clustering’, is one of the most well-studied problems in the field of network science [7, 13, 15, 19, 21]. Generally, the goal of community detection algorithms is to partition a graph into disjoint or overlapping [31] vertex partitions. While the exact optimization metric for this partitioning varies, one generally attempts to group vertices with similar attributes into the same community membership. In the absence of metadata, labels, or other non-topological attributes, basic network topology is used as the defining metric for ‘similarity’. In this case, the measure of network modularity is a natural optimization metric.

Modularity can be loosely defined as to how *well* a given network topology is divided into a given set of communities relative to a random network [26]. It is an important heuristic for community detection, and it has played a significant role in the literature [6, 10, 15, 18]. For a given set of communities, modularity is defined as the difference between the number of observed and number of expected intra-community edges in a network. If the modularity is high, the former is larger than the latter, and the given communities are considered ‘more

clustered than expected'. For a given graph $G = (V, E)$, and defined labels for each node $\{c_i\}_{i \in V}$, modularity is explicitly computed as in Eq. 1.

$$Q = \frac{1}{2m} \sum_{u,v \in V} \left[A_{uv} - \frac{d_u d_v}{2m} \right] \delta(c_u, c_v) \quad (1)$$

Here A is the adjacency matrix for the graph G , m is the number of edges in the network, d_u, d_v are the degrees of a nodes $u, v \in V$, and $\delta(\cdot, \cdot)$ is the Kronecker delta function. Despite how notable modularity maximization is in literature, it has some known problems such as the ‘resolution limit’ [14,22] in which the method cannot discern communities under a certain size for a given network density. A far less discussed issue with the modularity heuristic is its dependence on the Chung-Lu graph model [8,30], which assumes a random wiring of edges for a given degree distribution in expectation with self-loops and multi-edges. In essence, computing modularity is simply evaluating the null hypothesis on a given community assignment and network topology with one on a random null model graph provided by Chung-Lu. The contribution of the model comes from the term $\frac{d_u d_v}{2m}$ in Eq. 1. There are a significant number of ways to define a null model within a discussion on network topology. An excellent recent review article by Fosdick et al. [16] discusses the many considerations. Changing the implicit null model in modularity maximization has also been shown to produce substantial difference in the obtained communities in the case of geometrically constrained graphs [20].

In our prior work [2,3,17] and other related work [4,11,12,27], the general issues with the usage of Chung-Lu probabilities were studied for random graph generation. These problems include the theoretical reality that Chung-Lu generation cannot actually produce a vast majority of possible degree distributions [3], and the considerable error that results when using Chung-Lu probabilities for simple graph generation [4]. Similar issues arise when using Chung-Lu probabilities as an implicit null model in methods such as modularity maximization¹. Fosdick et al. [16] was the first work we know of in the literature that gave such considerations more than a cursory glance.

Hence, the primary focus of this work is to experimentally examine the impact of using proper null graph probabilities in place of the Chung-Lu model for the specific problem of community detection via modularity maximization. We additionally utilize methods from our prior work [2,3], which were in part motivated by the Fosdick et al. review, to derive vertex pairwise attachment probabilities. These attachment probabilities are then used in place of Chung-Lu probabilities within the computation of modularity for a maximization algorithm.

Modularity maximization [6,10,18,26] methods generally take two forms – single or multi-level methods. Single level methods make choices on vertex–community membership for each individual vertex, while multi-level methods

¹ One of the major issues is that a majority of graphs studied in community detection fall squarely in the simple graph space. We note this applies to LFR and similar benchmark graphs and a large proportion of the real-world graphs with defined communities; e.g., those listed in the SNAP repository.

assign vertices to communities and then iteratively coarsen communities into single vertices to improve computation time. We will specifically consider the former category of algorithms in this paper, although we will also discuss how our methods can be extended to multi-level algorithms such as the popular Louvain algorithm [1].

Generally, the outputs of community detection algorithms on networks are evaluated in two primary ways [7, 21]. If some *ground truth* community definitions exists for the network, the output of the algorithm is compared to this ground truth via computing a metric such as normalized mutual information (NMI). Absent an explicit ground truth, evaluation can be done by comparing some computed metric derived solely from the community assignments and network structure. In the latter case, modularity is one such popular measure. However, in the context of this paper, we are unable to directly use modularity scores, since our experimental differences are in *the way we compute modularity*. Hence, our experimental evaluations focus on comparisons to network ground truth with the NMI metric. To generate a suitably large number and variety on test instances, we utilize the common Lancichinetti-Fortunato-Radicchi (LFR) benchmark generator [23]. We will discuss our experimental setup in more detail later in this manuscript.

In summary, our contributions are as follows:

1. We are the first work to extensively study the usage of a more appropriate null graph model within the context of modularity maximization.
2. We detail our approach to computing attachment probabilities and their effective utilization within a modularity maximization framework.
3. We observe that this change in attachment probabilities can improve computed NMI scores by up to fifty percent on average for some data sets.
4. We discuss how this work might be applied in future efforts, such as with multi-level community detection algorithms.

2 Methods

Ultimately, including custom probabilities into the computation of modularity is trivial once one determines these probabilities. All one needs to do is replace the Chung-Lu term ‘ $\frac{d_u d_v}{2m}$ ’ with a general term for the connection probability ‘ p_{uv} ’ between nodes u and v as in Eq. 2. Algorithmically, however, there are some unique considerations which have to be made when using custom null models for modularity maximization.

$$Q = \frac{1}{2m} \sum_{u,v \in V} [A_{uv} - p_{uv}] \delta(c_u, c_v) \quad (2)$$

When the modularity maximization algorithm does not affect the underlying structure of the network, one can use a straightforward modularity maximization algorithm such as Clauset-Newman-Moore’s algorithm [9]. As such, we implemented this algorithm to utilize any arbitrary p_{uv} and validated it against

NetworkX’s implementation² with Chung-Lu probabilities. However, for multi-level methods such as the well-established Louvain maximization algorithm [1], null model discovery would need to be performed at every step. We will discuss future work intended to address this current drawback. Below, we will describe two methods for determining the p_{uv} attachment probabilities of the underlying null model of an arbitrary graph.

Algorithm 1. Computing double edge swap probabilities.

```

1: procedure DESSAMPLE( $G = (V, E), k$ )
2:    $P \leftarrow \frac{1}{k}$  get_adjacency( $G$ )
3:    $E' \leftarrow$  permute( $E$ )
4:   for  $i \in [0, \dots, k - 1]$  do
5:     for  $j \in [0, \dots, |E| - 1]$  do
6:       if valid_swap( $G, E_j, E'_j$ ) then
7:          $G \leftarrow$  perform_swap( $G, E_j, E'_j$ )
8:    $P \leftarrow P + \frac{1}{k}$  get_adjacency( $G$ )

```

Our first method, termed *desmod*, uses *double edge swaps* to randomly alter the graph topology while keeping the degree sequence the same. Double edge swaps take two edges and ‘swap’ the endpoints of each edge, permuting the edge-list of the graph while keeping the degree distribution consistent. The main idea behind the *desmod* method is to randomly sample a large number of realized instances of graphs with a fixed degree distribution to discern average degree-degree pairwise connection probabilities. We utilize a sampling method modified from our prior work [17, 28], which is based on established techniques [16], [?]. This method, as outlined in the Algorithm 1, involves first permuting the edge-list and matching each edge in the permuted list with an edge in the original list. These will be our potential double edge swap partners. Of course, some of these swaps will not be *feasible* (i.e., results in a multi-edge or self loop), so we check each potential swap for viability using the primitive `valid_swap()`. For more details on this method and how valid swaps are chosen, we refer the reader to Fosdick et al. [16] and our prior work Garbus et al. [17].

Our second method, termed *mlemod*, is based on our recent prior work [2], which shows that degree distributions can be better approximated by Chung-Lu like methods with connection probabilities determined by maximum likelihood estimation. For detailed explanations of how the probabilities are computed, please see the referenced manuscript. The overarching idea behind the algorithm is that a degree sequence can be seen as a probability distribution, and the degrees of nodes with common weights will be distributed as Poisson distributions. To match this probability distribution, and hence the degree sequence we want, we can use maximum likelihood estimation to express this distribution as a sum of Poisson distributions from which nodal weights can be discerned

² Implemented via the `greedy_modularity_communities()` function.

Algorithm 2. Computing maximum likelihood probabilities.

```

1: procedure MLEPROBS( $G = (V, E)$ )
2:    $D \leftarrow \text{degrees}(G)$ 
3:    $w \leftarrow \text{mle\_weights}(D)$ 
4:    $D \leftarrow \text{sort}(D)$ 
5:    $w \leftarrow \text{sort}(w)$ 
6:    $P \leftarrow \text{zeros}(|V|, |V|)$ 
7:    $m \leftarrow \frac{1}{2} \text{sum}(w)$ 
8:   for  $u \in V$  do
9:     for  $v \in V$  do
10:       $P_{uv} \leftarrow \frac{w_u w_v}{\sum_{x \in V} w_x}$ 

```

for a Chung-Lu-like graph model. The *mlemod* method has some unique issues to be considered when compared with *desmod*. With *desmod*, each probability is assigned explicitly to an edge, meaning that the probabilities output can be directly applied to perform modularity maximization using Eq. 2. In *mlemod* we only obtain a list of weights, from which probabilities can be derived once they are assigned to explicit nodes. This means that *mlemod* requires us to assign each node the ‘most probable’ label.

For a given graph $G = (V, E)$, with $|V| = n$, and weights returned from a process such as MLE $w = \{w_1, \dots, w_n\}$, the labeling problem amounts to finding a bijection $\phi : V \rightarrow w$ such that the probability of observing the given edge set $P(E|\phi(V))$ is maximized. If we call the sum of weights $\sum_{z \in V} \phi(z) = S$ we get Eq. 5.

$$P(E|\phi(V)) = \left[\prod_{uv \in E} P(e_{uv}|\phi(u), \phi(v)) \right] \left[\prod_{xy \notin E} (1 - P(e_{xy}|\phi(x), \phi(y))) \right] \quad (3)$$

$$= \left[\prod_{uv \in E} \frac{\phi(u)\phi(v)}{S} \right] \left[\prod_{xy \notin E} \left(1 - \frac{\phi(x)\phi(y)}{S} \right) \right] \quad (4)$$

$$\propto \left[\prod_{uv \in E} \phi(u)\phi(v) \right] \left[\prod_{xy \notin E} (S - \phi(x)\phi(y)) \right] \quad (5)$$

Broadly speaking, the above equation states the following. The optimal way to match weights with nodes is to maximize the product of weights across existing edges, while minimizing the product of weights among non-existent edges. In general, this is a difficult optimization problem. However, we can heuristically come up with an easy approximate solution by only considering the leftmost term, which corresponds to maximizing the likelihood of existing edges.

$$OBJ = \max_{\phi} \left[\prod_{uv \in E} \phi(u)\phi(v) \right] = \max_{\phi} \left[\prod_{u \in V} \phi(u)^{d_u} \right] \quad (6)$$

In this case, we have the objective outlined in Eq. 6. Fortunately, this is a very easy to maximize objective, as it only requires sorting the degrees of our graph in descending order and matching them with the associated weights, also sorted in descending order. This process can be observed in Algorithm 2.

3 Results

Our experiments were run on the server *Bella* at RPI. *Bella* has $2 \times$ AMD Epyc 7742 processors with 64 cores at 2.25 GHz and 2 TB DDR4 at 2666 MHz, and it is running Ubuntu 20.04.6 with Python version 3.8.10 and NetworkX version 3.1.

We ran four sets of tests with various topological differences. For these experiments, all graphs were generated using the `LFR_benchmark_graph()` function available in Python’s `NetworkX` library. A range of parameters were input, and instances in which the generator failed regarding combinations of those parameters were skipped. The definitions for the various adjusted input parameters are given in Table 1. We compare community outputs using NMI values as well as the number of communities generated in comparison to the ground truth. We performed two primary experiments, which we will define below.

Table 1. Variable Definitions for Experimental LFR Generation.

Variable	Definition
N	Number of nodes in the network
$\langle k \rangle$	Average degree of nodes in the network
k_{max}	Maximum degree of nodes in the network
k_{min}	Minimum degree of nodes in the network
s_{min}	Minimum community size in the network
τ_1	Power-law exponent of node degree distribution
τ_2	Power-law exponent of community size distribution
μ	Average proportion of edges that are external to communities

3.1 General Experimental Set of Networks

We tested the results for a general set of graphs with a lower bound on community size. For these tests, we generated LFR’s with the following parameters: $N \in \{4000, 8000, 16000\}$, $\tau_1 \in \{2, 2.5\}$, $\tau_2 \in \{1.1, 1.25, 1.5\}$, $\mu \in \{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$, $s_{min} = 6$, $k_{min} = 5$, and $k_{max} \in \{\frac{N}{20}, \frac{6N}{100}, \frac{7N}{100}, \frac{8N}{100}, \frac{9N}{100}, \frac{N}{10}\}$. The NMI comparison between modularity maximization via the probabilities of Chung-Lu (`chung`), `desmod` (`samp`), and `mlemod` (`mle`) can be observed in Fig. 1. We also show the number of communities obtained by each of these three methods plus the ground truths for this dataset in Fig. 2. We show this due to possible ‘NMI hacking’ which can occur for randomly chosen communities as described in work by Vinh et al. [29] where higher

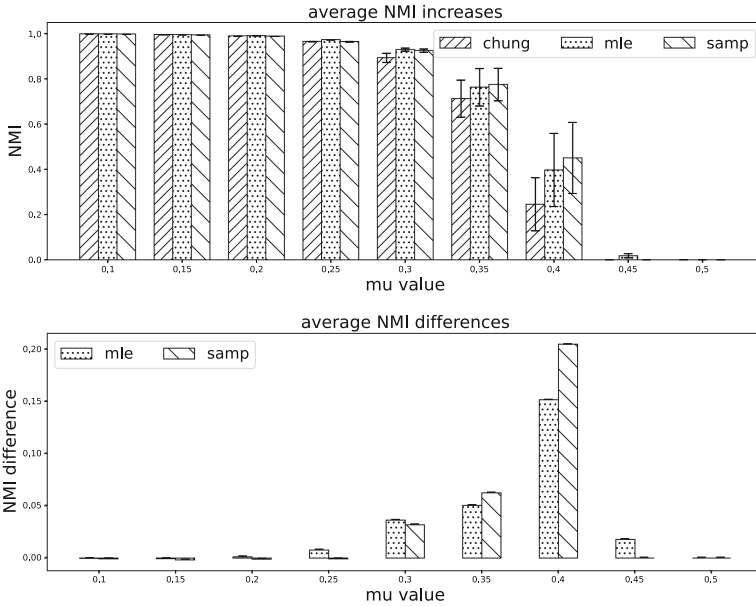


Fig. 1. NMI for general test set: NMI results for standard modularity maximization, *desmod*, and *mlemod* varying the μ parameter on LFR graphs with minimum degree 5, and minimum community size 6. We can see a sharp improvement in cluster quality near the $\mu = 0.4$ bound, implying that *desmod* and *mlemod* may perform better than standard modularity maximization in these test instances.

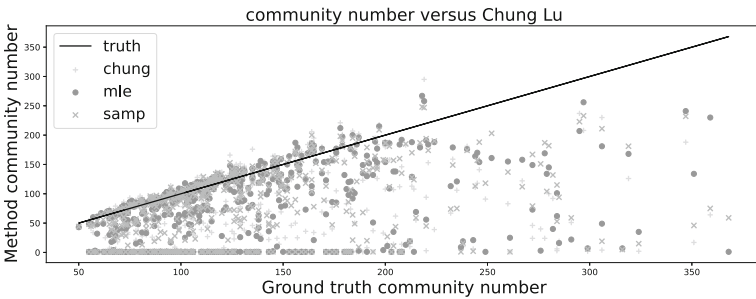


Fig. 2. Community numbers for general tests: The number of communities output by various methods in comparison to the ground truth communities. Each point is a specific graph, and the marker denotes the method used to obtain that graphs clustering. We can see that in most cases the number of communities is near, or below the expected number.

numbers of communities may sometimes yield higher NMI values without necessarily being better at matching the ground truth communities.

As Fig. 1 demonstrates, our proposed null model choices both improve upon the NMI computed on outputs using baseline Chung-Lu probabilities, on average.

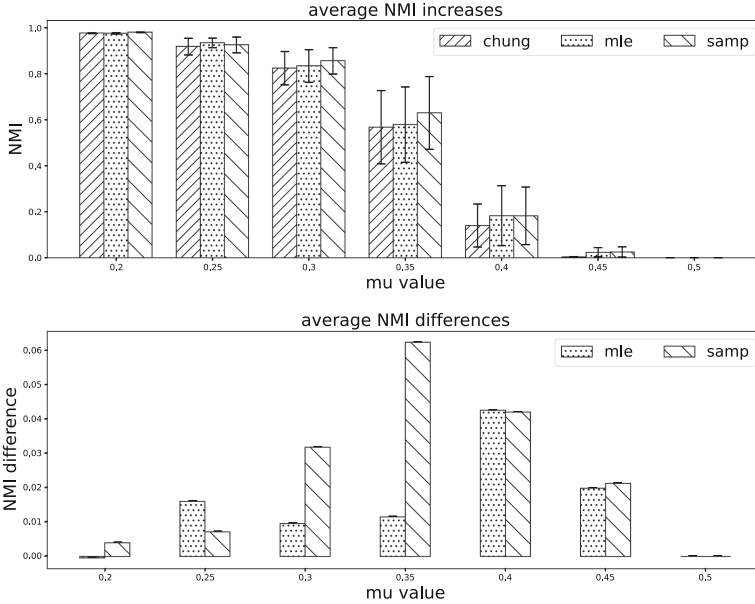


Fig. 3. NMI for $\tau_2 = 1.1$: NMI results for a narrow band of LFR graphs with $\tau_2 = 1.1$. Note that both *mlemod* and *desmod* have varying behavior in their NMI-difference plots in comparison to Fig. 1. Regardless, we can still see that both *mlemod* and *desmod* outperform standard modularity maximization for every test instance, on average.

This difference is most notable at μ values approaching $\mu = 0.5$, the point at which communities become difficult to discern for all methods. This is expected, as $\mu = 0.5$ implies that the each node on average has as many edges external to their ground truth community as they have internal edges. We also note that our outputs closely match the ground truth in terms of the number of communities, as shown in Fig. 2. Generally, our proposed null models either closely match the ground truth, or it results in fewer communities output. This is possibly a consequence of the resolution limit, where multi-level methods, described later, would be able to improve upon these results in future work.

3.2 Fixed Community Size Distribution Experimental Set

Additionally, we explored the quality of clusters for LFR graphs with a fixed τ_2 parameter. This can be seen in Fig. 3. We generated LFR's with the following parameters: $N \in \{1000, 2000, 4000, 8000\}$, $\mu \in \{0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$, $\tau_1 \in \{2, 2.5\}$, $\tau_2 \in \{1.1\}$ ³, $\langle k \rangle \in \{20, 25, 30, 35\}$, and $k_{max} \in \{100, 150, 200, 250, 300\}$. The goal for these experiments was to observe how the methods perform as the distribution of clusters remains close to linear, which we noticed made community detection much more ‘difficult’ for our modularity maximization meth-

³ We chose $\tau_2 = 1.1$, as NetworkX failed to generate graphs using $\tau_2 = 1.0$.

ods. This makes these graphs topologically unique among the generated LFR graphs, since their largest and smallest communities vary far less drastically.

We give the results of this experimental set in Fig. 3, again as a comparison between output NMI values. As shown in Fig. 3, we once more observe that *desmod* (samp) and *mlemod* (mle) result in consistently higher NMI scores than standard Chung-Lu (chung) probabilities. However, in these tests, we find that the output NMI takes a different form to the general case. Here, the difference in NMI can be seen to peak at $\mu = 0.35$, and only for *desmod*. This implies that sampling may perform better under certain topological features. Meanwhile, *mlemod* under-performs in comparison to *desmod* for many μ values, implying that it too may suffer as a consequence of certain topological properties.

4 Discussion

The results in this manuscript indicate that the choice of null model has a significant impact on the observed cluster quality for many graphs. We find that the results for Chung-Lu, sampled (*desmod*), and MLE probabilities (*mlemod*) are consistently better for general LFR graphs with broadly-varying parameters and size. On average, across a wide range of tests, sampled and MLE probabilities achieve better NMI results than those of standard modularity maximization. In the particular cases of larger LFR graphs and LFR graphs with smaller τ_2 values, both methods consistently out-perform Chung-Lu based modularity maximization. The differences are particularly notable near the $\mu = 0.4$ boundary.

The observed results suggest that *mlemod* and *desmod* outperform standard modularity maximization in general, particularly when the connectivity between communities is relatively high. This implies these null models are more robust to network density than the Chung-Lu random graph model. The primary question the reader might have is: **Why?**

The explanation can actually be summarized quite succinctly. In our prior work [17], we have observed that Chung-Lu probabilities can over-estimate real attachment probabilities⁴ between pairs of average degree nodes and pairs of high degree nodes within graphs with skewed degree distributions; low degree probabilities are otherwise similar. As a consequence, the baseline modularity maximization biases *against* assortativity, while most real networks and benchmark networks actually exhibit a considerable amount of assortative degree mixing within communities [5, 24, 25]. The use of appropriate null model probabilities ‘re-biases’ towards assortative mixing within communities when performing modularity maximization.

We finally note one specific concern that may arise to the reader, in that our modularity maximization method used for experimentation is relatively naive compared to more modern modularity maximization algorithms. In the following subsection, we provide a theoretical justification for how one may extend a multi-level Louvain-type algorithm for use with our proposed null models.

⁴ Here, ‘real attachment probabilities’ are those determined for an appropriate simple graph null model under an appropriate sampling methodology.

4.1 Extension to Explicit Multi-level Methods

One issue that arises when using bespoke null models defined by methods such as maximum likelihood estimation or sampling is that it excludes simple implementations of explicit multi-level schemes or approaches that otherwise modify the underlying network structure before maximizing modularity. E.g., in standard Louvain methods for modularity maximization, small communities are found on the original graph according to a greedy approach such as the one that we use in this paper. These communities are then coarsened into a single vertex in a weighted graph or multigraph, where modularity maximization is again performed. This allows for information from multiple scales to be considered, and it leads to generally better results. Louvain and many other more modern algorithms using this, and related approaches appear throughout the literature [7, 19].

Regardless, multi-level schemes may be implemented using custom graph-specific null models if we allow for extra computational overhead. Consider maximizing modularity the same way it is currently done in this paper. Then, a coarsened multi-graph may still be obtained, as in Louvain. Because we are not using Chung-Lu probabilities in this case, we do not have straightforward access to a null model for the coarsened graph. However, we can consider the probability of connection between the communities given by our initial clustering. In this case, the probability of connection between two clustered ‘supernodes’ can be thought of as the sum of connection probabilities between their comprising nodes. This can be seen in Eq. 7, where C_i is a community of nodes given by the first round of modularity maximization.

We check that this is consistent with the behavior we expect from the Chung-Lu random graph model. This is shown in Eq. 8, where we show that using Chung-Lu probabilities in accordance to the prior Eq. 7 yields the associated Chung-Lu probabilities of the coarsened graph.

$$p_{C_i C_j} = \sum_{u \in C_i} \sum_{v \in C_j} p_{uv} \quad (7)$$

$$= \frac{1}{2m} \sum_{u \in C_i} \sum_{v \in C_j} d_u d_v = \frac{|C_j|}{2m} \sum_{u \in C_i} d_u \langle d_{C_j} \rangle = \frac{d_{C_i} d_{C_j}}{2m} \quad (8)$$

While this suggests the proposed method may be consistent with that of Chung-Lu, experimental validation should be performed to ensure that this provides meaningful communities. This is a primary topic of investigation for our ongoing and future work.

5 Conclusion

In this manuscript, we investigated the effects of null model choice on modularity maximization. We did this through a series of experiments using the LFR benchmark with varying parameters. We developed two different methods for altering modularity maximization for custom graph null models. One method, which we

call *desmod*, uses double edge swaps on the input graph to generate probabilities. The other method, which we call *mlemod*, uses a maximum likelihood method along-side a greedy labeling to determine attachment probabilities. By replacing standard modularity maximization with *desmod* and *mlemod*, we found that the normalized mutual information of output communities relative to the ground truth was better in many cases. In particular, both *mlemod* and *desmod* yield better results than Chung-Lu methods when inter-community connections are dense. We additionally suggest that this technique may be extended to the case of multi-level schemes such as Louvain, and we provide some theoretical justification for how that may be done. Despite this, such studies are left for future work and remain an open problem. This work represents a ‘first foray’ into the practical use of null model choice for community detection on general graphs, and it is the authors’ hope that this spurs an interest in the topic for the broader community.

References

1. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech: Theory Exp.* **2008**(10), P10008 (2008)
2. Brissette, C., Liu, D., Slota, G.M.: Correcting output degree sequences in Chung-Lu random graph generation. In: Cherifi, H., Mantegna, R.N., Rocha, L.M., Cherifi, C., Micciche, S. (eds.) *COMPLEX NETWORKS 2016 2022*, vol. 1078, pp. 69–80. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-21131-7_6
3. Brissette, C., Slota, G.M.: Limitations of Chung Lu random graph generation. In: Benito, R.M., Cherifi, C., Cherifi, H., Moro, E., Rocha, L.M., Sales-Pardo, M. (eds.) *COMPLEX NETWORKS 2021*, vol. 1072, pp. 451–462. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-93409-5_38
4. Britton, T., Deijfen, M., Martin-Löf, A.: Generating simple random graphs with prescribed degree distribution. *J. Stat. Phys.* **124**(6), 1377–1397 (2006)
5. Catanzaro, M., Caldarelli, G., Pietronero, L.: Assortative model for social networks. *Phys. Rev. E* **70**(3), 037101 (2004)
6. Chen, M., Kuzmin, K., Szymanski, B.K.: Community detection via maximization of modularity and its variants. *IEEE Trans. Comput. Soc. Syst.* **1**(1), 46–65 (2014)
7. Chunaev, P.: Community detection in node-attributed social networks: a survey. *Comput. Sci. Rev.* **37**, 100286 (2020)
8. Chung, F., Lu, L.: The average distances in random graphs with given expected degrees. *Proc. Natl. Acad. Sci.* **99**(25), 15879–15882 (2002)
9. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. *Phys. Rev. E* **70**(6), 066111 (2004)
10. Despalatović, L., Vojković, T., Vukicević, D.: Community structure in networks: Girvan-newman algorithm improvement. In: 2014 37th international convention on information and communication technology, electronics and microelectronics (MIPRO), pp. 997–1002. IEEE (2014)
11. Drobyshevskiy, M., Turdakov, D.: Random graph modeling: a survey of the concepts. *ACM Comput. Surv. (CSUR)* **52**(6), 1–36 (2019)
12. Durak, N., Kolda, T.G., Pinar, A., Seshadhri, C.: A scalable null model for directed graphs matching all degree distributions: in, out, and reciprocal. In: 2013 IEEE 2nd Network Science Workshop (NSW), pp. 23–30. IEEE (2013)

13. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3–5), 75–174 (2010)
14. Fortunato, S., Barthelemy, M.: Resolution limit in community detection. *Proc. Natl. Acad. Sci.* **104**(1), 36–41 (2007)
15. Fortunato, S., Hric, D.: Community detection in networks: a user guide. *Phys. Rep.* **659**, 1–44 (2016)
16. Fosdick, B.K., Larremore, D.B., Nishimura, J., Ugander, J.: Configuring random graph models with fixed degree sequences. *SIAM Rev.* **60**(2), 315–355 (2018)
17. Garbus, J., Brissette, C., Slota, G.M.: Parallel generation of simple null graph models. In: *The 5th IEEE Workshop on Parallel and Distributed Processing for Computational Social Systems (ParSocial)* (2020)
18. Good, B.H., De Montjoye, Y.A., Clauset, A.: Performance of modularity maximization in practical contexts. *Phys. Rev. E* **81**(4), 046106 (2010)
19. Jin, D., et al.: A survey of community detection approaches: from statistical modeling to deep learning. *IEEE Trans. Knowl. Data Eng.* **35**(10), 1149–1170 (2021)
20. Kishore, R., Gogineni, A.K., Nussinov, Z., Sahu, K.K.: A nature inspired modularity function for unsupervised learning involving spatially embedded networks. *Sci. Rep.* **9**(1), 2631 (2019)
21. Lancichinetti, A., Fortunato, S.: Community detection algorithms: a comparative analysis. *Phys. Rev. E* **80**(5), 056117 (2009)
22. Lancichinetti, A., Fortunato, S.: Limits of modularity maximization in community detection. *Phys. Rev. E* **84**(6), 066122 (2011)
23. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**(4), 046110 (2008)
24. Newman, M.E.: Assortative mixing in networks. *Phys. Rev. Lett.* **89**(20), 208701 (2002)
25. Newman, M.E.: Mixing patterns in networks. *Phys. Rev. E* **67**(2), 026126 (2003)
26. Newman, M.E.: Modularity and community structure in networks. *Proc. Natl. Acad. Sci.* **103**(23), 8577–8582 (2006)
27. Slota, G.M., Berry, J., Hammond, S.D., Olivier, S., Phillips, C., Rajamanickam, S.: Scalable generation of graphs for benchmarking HPC community-detection algorithms. In: *IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC)* (2019)
28. Slota, G.M., Garbus, J.: A parallel LFR-like benchmark for evaluating community detection algorithms. In: *The 5th IEEE Workshop on Parallel and Distributed Processing for Computational Social Systems (ParSocial)* (2020)
29. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: is a correction for chance necessary? In: *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1073–1080 (2009)
30. Winlaw, M., DeSterck, H., Sanders, G.: An in-depth analysis of the Chung-Lu model. Technical report, Lawrence Livermore National Lab. (LLNL), Livermore, CA (United States) (2015)
31. Xie, J., Kelley, S., Szymanski, B.K.: Overlapping community detection in networks: the state-of-the-art and comparative study. *ACM Computing Surveys (CSUR)* **45**(4), 1–35 (2013)



On Centrality and Core in Weighted and Unweighted Air Transport Component Structures

Issa Moussa Diop¹(✉), Cherif Diallo¹, Chantal Cherifi², and Hocine Cherifi³

¹ LACCA, Gaston Berger University, Saint-Louis, Senegal
diop.issa-moussa@ugb.edu.sn

² DISP LAB, University of Lyon 2, Lyon, France

³ LIB, University of Burgundy, Dijon, France

Abstract. This study explores the mesoscopic intricacies governing the interplay of regional and interregional traffic flows within the world air transportation network. We adopt a network decomposition method, the component structure, to partition the network into local and global components, aligning with geographical and cultural boundaries. We explore the centrality and the core of these unweighted and weighted networks. The analysis of centrality highlights distinctions between airports with high traffic and those with high degrees. This contrast is particularly pronounced in the global air network and its largest component. Additionally, core analysis reveals similarities between the *s*-core and the *k*-core for local and global components, despite the latter encompassing more airports. In the world air network, North and Central America-Caribbean airports dominate the *s*-core, while European airports take precedence in the *k*-core. Interestingly, airports distant from their component's core can also rank among the top high-strength airports. This comparative analysis underscores the component structure representation's vital role in uncovering regional and interregional dynamics within the world air transportation network, providing valuable insights into its mesoscopic structure and implications for global air travel.

Moreover, we explore the strength and the *s*-core of these networks. Results display fewer local components well delimited and more global components covering the world than the unweighted world air transportation network. Centrality analysis reveals the difference between the top airports with high traffic and the top airports with high degrees. This difference is more pronounced in the global air network and the largest global component. Core analysis shows similitude between the *s*-core and the *k*-core for the local and global components, even though the latter includes more airports. For the world air network, the North and Central America-Caribbean airports dominate the *s*-core, whereas the European airports dominate the *k*-core. Transport network and its intricate hierarchical organization.

Keywords: weighted networks · unweighted network · mesoscopic structure · component structure · World Air Transportation Network · centrality · core

1 Introduction

The intricate and ever-evolving web of global air transportation networks is pivotal in connecting distant regions, fostering economic growth, and shaping our interconnected world. As the backbone of international travel and trade, this intricate network comprises a vast array of airports, routes, and connections that collectively serve as conduits for the movement of people and goods. To comprehend this colossal system's underlying structure and dynamics, it is imperative to delve beyond the macroscopic view and explore the mesoscopic intricacies that govern the interplay of regional and interregional traffic flows.

Understanding the mesoscopic structure of complex networks has been a subject of significant research interest [1–9]. Several studies have offered valuable insights into the organization and dynamics of such networks, including transportation systems [10, 11].

[12] study air transportation networks, finding communities aligned with geography. They defined seven city roles based on link patterns: Provincial hubs, Connector hubs, Kinless hubs, Ultrapерipheral nodes, Peripheral nodes, Nonhub connectors, and Nonhub kinless. This work highlights geopolitical influences on network formation and city roles.

[13] divide the global air transportation network into seven regions (Africa, Asia, Europe, Latin America, Middle East, North America, Southwest Pacific) using OAG data. They find small-world networks with a two-regime power-law degree distribution. Their targeted attack experiments reveal that networks with larger cores are more resilient than those with smaller cores.

To examine the relationships within the unweighted global air transportation network and its regional constituents, we employ a network decomposition method known as the “component structure,” as outlined in [14]. This decomposition technique partitions the network into two fundamental elements: local components and global components. Local components represent densely connected regions within the original network, while the interconnections between these local components, along with their respective nodes, constitute the global components. This approach allows us to delineate the world's air transportation system into distinct regional components that align with geographical and cultural boundaries.

In contrast, the weighted air transportation network is not much studied. Indeed, in [15], the Australian airport network exhibits strength-degree correlation, with Sydney as a major hub, and weighted clustering lower than unweighted, indicating topological clustering. In [16], Asian passenger aviation in 2014 and 2018 highlights Low-Cost Carriers' growth and centrality measures ranking Changi, Incheon, Narita, and Hong Kong airports highly. In [17], the weighted world air transportation network demonstrates heavy-tailed degree and strength distributions, a strength-degree linear relationship, a rich-club phenomenon, and assortative degree-degree correlation.

The articles considering the mesoscopic structure do not consider the weighted aspect. Vice versa, papers that examined the weighted aerial networks did not consider their mesoscopic structure. This paper conducts a comprehensive comparative examination of the route-based (unweighted) and traffic-based (weighted) networks, delving into mesoscopic and microscopic levels to gain deeper insights into the distinctions between the structure and traffic of the global air transportation network. We investigate and compare the k -core and s -core structures at the mesoscopic level. Lastly, we

assess and differentiate between node degree and strength properties at the microscopic level.

The world air transport network comprises 2,734 nodes and 16,665 links, sourced from FlightAware [18], encompassing flights between May 17, 2018, and May 22, 2018 [19]. In this representation, nodes correspond to airports, while links denote direct flights connecting these airports. The volume of flights between each pair of airports determines the link weights.

The remainder of this paper is structured as follows: Sect. 2 delves into the findings of the core structure analysis, while Sect. 3 provides the outcomes of the comparative analysis concerning the strength centrality of the components and the global air transportation network. Lastly, our conclusions are summarized in Sect. 4.

2 Core Structure Analysis

This section reports the Max s -core analysis of the large weighted components and a comparative investigation with their corresponding Max k -core. Additionally, it presents a similar analysis of the world transportation network.

The graph's k -core [20] is the subgraph obtained by recursively removing all the vertices of degree smaller than k until the degree of all remaining vertices is larger than or equal to k . By extension, the s -core of a weighted graph is a sub-network in which a node has at least a strength s . One can extract the maximum s -core by removing nodes iteratively from the network. Indeed, the $s_{\min(s_i)}$ -core, where each node has at least a strength 1, is the whole network. One forms the next level by removing all the nodes with the minimum strength $s_{\min(s_{i+1})}$. The remaining nodes form $s_{\min(s_{i+1})}$ -core, and so on until one reaches the core number Max s_n -core for which it is impossible to obtain the $s_{\min(s_{n+1})}$ -core.

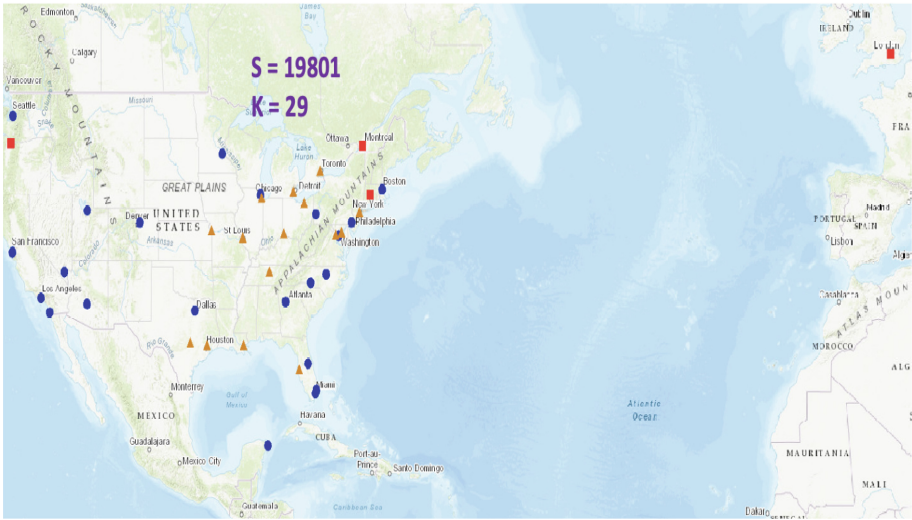
2.1 Local Components

Figure 1 reports the maximum s -core values of two weighted largest local components. It also shows the airports it includes.

The North and Central America-Caribbean component's Max s -core has 26 airports handling 22% of total traffic. On average, there are 1,238 flights between these airports. Three of them are outside the US: Montreal/P E Trudeau Airport in Canada, Cancun Airport in Mexico, and London Heathrow Airport in the UK, which has significant traffic in the US. Only Cancun Airport belongs to both the k -core and s -core. Four Max s -core airports (Raleigh-Durham, Pittsburgh, Windsor Locks, Cancun) serve within the s -core but aren't top 26 in traffic. While the Max s -core has fewer airports than the Max k -core (Fig. 1A), they share 22 airports (84.6% of Max s -core), indicating closely related traffic and destinations. However, airports like London Heathrow in the UK, Portland, Windsor Locks in the US, Montreal/Pierre Elliott Trudeau have more traffic than other Max s -core airports. Removed airports are mainly in the eastern US.

The Europe-Russia-Central Asia Max s -core comprises 27 airports, with 19.6% of the component's flights. On average, there are 632 flights between these airports, spanning multiple European countries. Pulvoko Airport is the sole Russian sub-region representative, while Ben Gurion Airport has the lowest traffic at 10,747 flights. John

A North and Central America - Carribean



B Europe – Russia - Central Asia

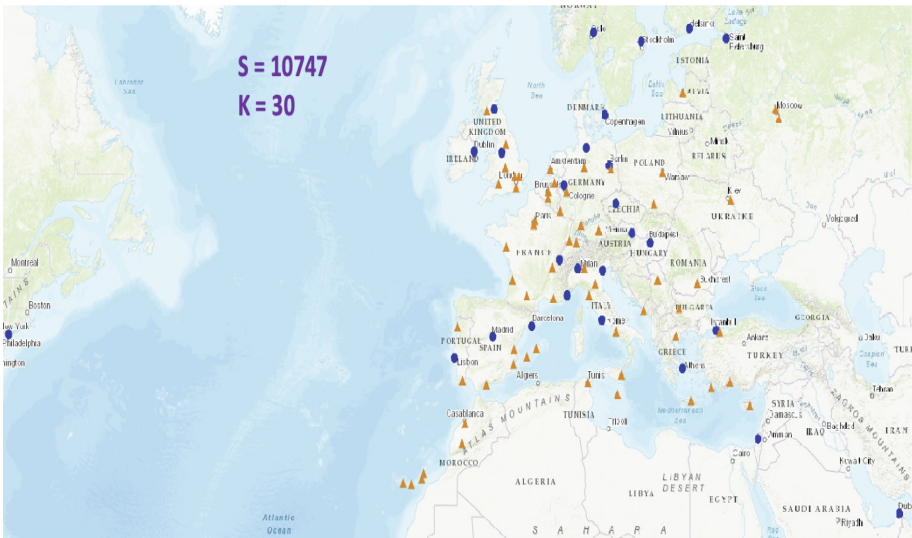


Fig. 1. The core of the weighted large local components consists of two key values: S , which represents the maximum s -core value, and K , which means the maximum k -core value. Blue points denote airports that belong to both the maximum s -core and the maximum k -core, red points indicate airports exclusively in the maximum s -core, and yellow points signify airports exclusive to the maximum k -core. For the North and Central America-Caribbean component, the maximum s -core encompasses 26 airports, while the Europe-Russia-Central Asia component has a maximum s -core comprising 27 airports.

F Kennedy in the USA handles the most flights within the s-core, serving as a vital Europe-US connection. Compared to the Max k-core, the Max s-core contains three times fewer airports, excluding all North African airports (Fig. 1B). In addition, several airports from France, Spain, Germany, and the UK. All Max s-core airports are part of the Max k-core, but countries like Germany, the UK, France, Italy, and Spain dominate the latter. Airports like Charles de Gaulle in France, Munich in Germany, Amsterdam in the Netherlands, and Zurich Airport in Switzerland are absent from the Max s-core, with traffic oriented more towards inter-regional destinations.

There are 18 airports in the East and Southeast Asia-Oceania Max s-core, serving different countries, mainly capitals and megacities. These airports handle 14% of the component's traffic, averaging 592 flights. Kansai Airport in Japan has the lowest traffic at 7,687 flights. Five airports (Brisbane Airport in Australia, Auckland Airport in New Zealand, Tan Son Nhat Airport in Vietnam, Ngurah Rai Airport in Indonesia, and Kansai Airport in Japan) in the Max s-core are absent from the top 18 strength airports. The Max s-core contains half as many airports as the k-core, distributed across various countries, with China dominating the Max k-core. Nine airports are common to both the s-core and the k-core, while several Chinese airports are absent from the s-core, replaced by nine airports from different countries, including Oceania.

The Max s-core of the Africa-Middle East-Southern Asia component includes nine airports, handling 20.1% of the traffic with an average of 659 flights between them. These airports are in three countries (5 in India, 2 in Saudi Arabia, and 1 in Germany), reflecting substantial traffic between Saudi Arabia and India due to the Indian workforce in Saudi Arabia. Rajiv Gandhi International Airport has the lowest traffic in the Max s-core at 4,609 flights. These airports form a complete graph and concentrate much of the component's traffic. All but Rajiv Gandhi Airport are among the top 9 strength airports. The Max k-core is twice as large and includes the Max s-core airports. Only Rajiv Gandhi Airport is part of the s-core, with nine airports from India and the Gulf countries absent. Nevertheless, India dominates both the Max s-core and the Max k-core.

Ten airports are in the Max s-core of the South America component. They are responsible for 24% of the component's traffic, with an average of 319 flights between them. These airports are all situated in Brazil, primarily along the country's east coast. Rio Galeão - Tom Jobim Airport has the lowest traffic, with 1,914 flights. Only two airports, Santa Genoveva Airport in Goiania and Deputado L E Magalhães Airport, do not appear among the top 10 strength airports. Importantly, all Max s-core airports are also part of the Max k-core, which is nearly twice the size. Brazil significantly influences both the Max s-core and the Max k-core.

In summary, two typical behaviors are observed for Max s-cores. In the first scenario, Max s-core airports are primarily located within a single country, such as the United States, India, and Brazil, which dominate their respective components' Max s-cores. Max s-core airports are more evenly distributed across the component in the second scenario. This pattern is seen in the East and Southeast Asia-Oceania and Europe-Russia-Central Asia components. Across all components, the Max s-cores handle approximately 20% of the traffic, except for East and Southeast Asia-Oceania, which accounts for 14%. The Max s-core consistently contains significantly fewer airports than the Max k-core, typically one-half to one-third. Moreover, most

Max s-core airports are also part of the Max k-core, indicating that airports with high traffic between them tend to have numerous connections. Indeed, the Max s-core generally includes only the most critical hubs, with exceptions in the Europe-Russia-Central Asia region, where some major hubs exhibit more inter-regional traffic.

2.2 Global Component

The Max s-core of the large global component presents 22 airports, spanning North America, Europe, and East and Southeast Asia. This s-core handles 22.19% of inter-regional flights, with Suvarnabhumi Airport in Thailand having the lowest traffic at 13,196 flights among these airports. Figure 2A illustrates the Max s-core of the large global component. Note that 72% of its airports also belonging to the k-core. A comparison between the s-core and the k-core reveals two trends. Firstly, in the Europe and East and Southeast Asia regions, we find nearly identical airports in both the s-core and k-core of the global component. Secondly, there are differences in the North America and East and Southeast Asia regions. In the former, four airports (General E L Logan Airport, Dallas Fort Worth Airport, Seattle Tacoma Airport, and Lester B. Pearson) are in the Max s-core but not in the Max k-core, while in the latter, all the airports in the k-core are absent from the s-core.

2.3 World Air Transportation Network

The Max s-core of the world air transportation network comprises 33 airports. North America accounts for 27 airports, with Europe and Japan contributing five and one airports, respectively. Minneapolis-St Paul Airport has the lowest traffic at 24,088 flights among these airports. Compared to the Max k-core, the Max s-core in the global air network contains fewer airports and a significant lack of overlap, as depicted in Fig. 2B. The Max k-core is notably more concentrated in Europe. These findings underscore the high traffic orientation of North American airports in the weighted world air transportation network, while European airports dominate the unweighted world air transportation network due to their extensive network of destinations. The component structure analysis reveals other significant airports worldwide, mitigating this disparity.

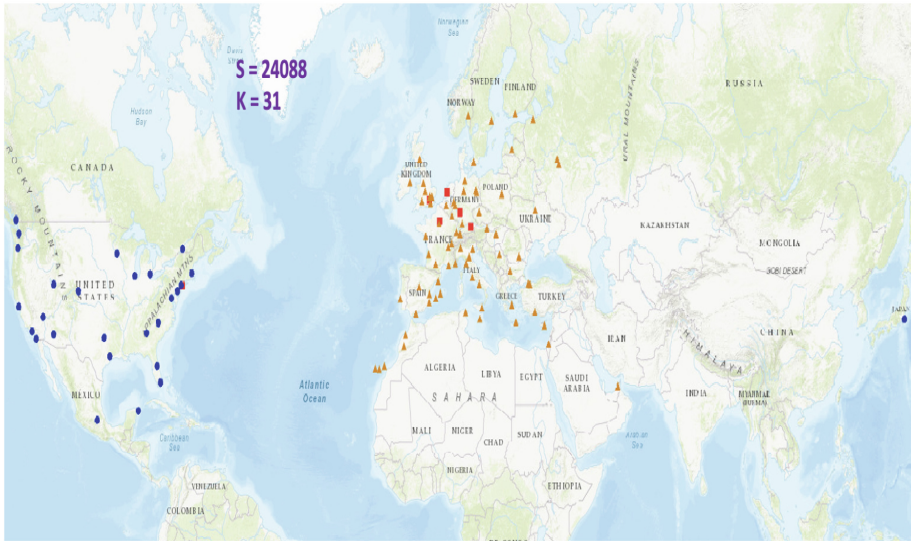
3 Local Topological Properties

Centrality analysis investigate the most influential nodes in a network. There are multiple definitions of centrality that exploit either local or global characteristics of the networks [21–25]. Here, we perform a comparative analysis of the Strength (Number of flights in an airport) and Degree (Number of routes in an airport) centralities of the various components. This analysis is in line with recent works considering the community structure to define new centrality measures [26–29].

3.1 Top Five Nodes Analysis

Local Components. This section analyzes the top five airports in descending order of the number of flights with airports in their local component (internal strength centrality).

A World air transportation network



B Global component

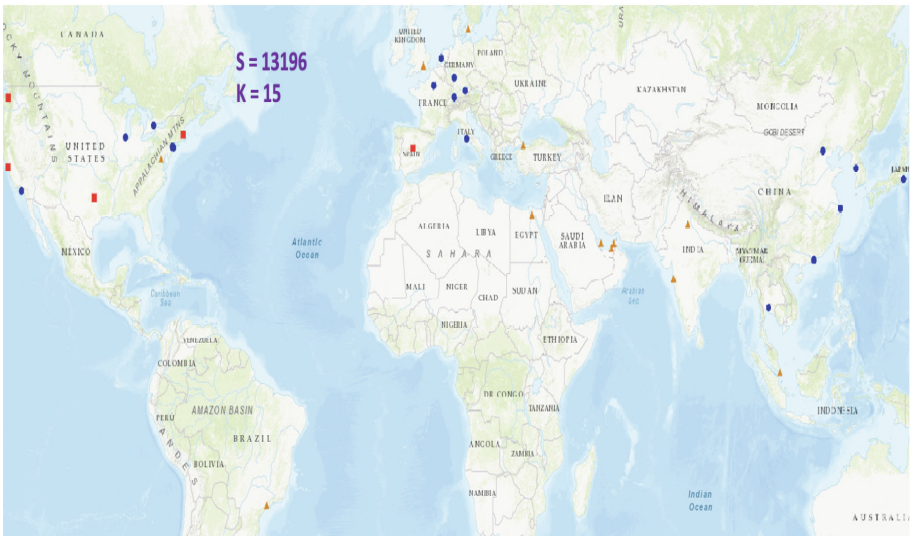


Fig. 2. A) There are 33 airports included in the maximum s-core of the weighted world air transportation network. B) The maximum s-core of the weighted large global component comprises 22 airports. In both cases, S represents the maximum s-core value, and K represents the maximum k-core value. Blue points indicate airports belonging to both the maximum s-core and maximum k-core, red points exclusively in the maximum s-core, and yellow points exclusively in the maximum k-core.

In the weighted *North and Central America-Caribbean component*, the top five airports remain in the United States, as in the unweighted network. Hartsfield-Jackson Atlanta Airport, Chicago O'Hare Airport, and Dallas/Fort Worth Airport are among the top five most connected airports. Los Angeles Airport ranks second in terms of flights, although it doesn't feature in the top airports for the number of routes. All these airports boast over 100 connections and handle more than 100,000 flights within this component. The fifth significant airport, Ronald Reagan Washington Airport, falls short of these numbers. Collectively, these airports manage nearly 15% of regional flights. It's worth mentioning that London Heathrow ranks ninth, operating around 74,000 flights. Denver and Houston airports, ranking among the top five for connections, place eleventh and sixteenth in terms of flights.

In the *Europe-Russia-Central Asia component*, the top five airports by traffic manage nearly 11% of the region's traffic. Only Amsterdam Schiphol Airport and Barcelona Airport are among the top five hubs by route. Ireland's Dublin Airport leads with over 4,700 flights. Charles de Gaulle Airport in France is the largest hub, ranking third in traffic. Surprisingly, John F. Kennedy Airport in the USA is fourth with 40 links, having more European flights than many local airports. Munich Airport and London Stansted Airport, both top hubs by routes, rank sixth and sixty-fifth by flights. Note that Frankfurt Airport, belongs to the *Africa-Middle East-Southern Asia component*. Russia's primary airport, Pulkovo Airport, is fourteenth. Dubai Airport, part of this component, is twenty-fifth, despite being the Middle East's largest airport.

In *East and Southeast Asia-Oceania*, China boasts three busiest airports, with Beijing Capital, Shanghai Pudong, and Guangzhou Baiyun ranking first, third, and fourth, respectively. Singapore Changi Airport takes second place, while Hong Kong Airport ranks fifth with fewer than 30,000 flights. China dominates both traffic and destinations in this region. Among the top five destinations, Chengdu Shuangliu Airport and Taiwan Taoyuan Airport stand at eleventh and ninth place, respectively. Sydney K Smith Airport, the major Oceania airport, secures the eighth spot in the rankings.

Indian airports dominate the *Africa-Middle East-Southern Asia region*. The top three busiest airports are in India, but not among the top five most connected. Saudi Arabia and Germany have two of the top five connected airports. Indira Gandhi Airport handles the most traffic with 20,000+ flights, followed by Chhatrapati Shivaji Airport. Kempegowda Airport ranks third, while King Abdulaziz is fourth. Frankfurt am Main Airport is fifth, with 30 internal connections and 11,000+ flights. Dubai Airport is the most connected in a different region, and Addis Ababa Bole Airport ranks nineteenth in flights.

Brazil dominates in the *South America region* with four of the top five airports, each with fewer than 8000 flights. Guarulhos G A F Montoro Airport stands out for both traffic and destinations. Tancredo Neves Airport in Belo Horizonte is first, El Dorado Airport in Colombia is third, Rio G-T Jobim Airport in Rio de Janeiro is fourth but less connected, and Presidente J Kubitschek Airport is fifth in Brazil's capital.

Global Component. We explore the top five airports participating in the inter-regional traffic in each large local component. These airports also belong to the large global component.

In North and Central America-Caribbean, four of the top five interregional traffic airports are in the USA, while one is in Canada. Chicago O'Hare is the exception among the top five busiest airports. John F. Kennedy and Newark Liberty, both in New York, lead the ranking, with JFK having double the traffic of others. These top three are also among the top 10 globally.

In Europe-Russia-Central Asia, Frankfurt, and Charles de Gaulle dominate inter-regional flights and destinations. Only Munich Airport is not among the top five most connected airports. These airports are all in the global top 10 and exclude Russian airports. European airports lead interregional flights.

In East and Southeast Asia-Oceania, none of the top five busiest interregional airports are in Oceania. Narita Airport in Japan is the busiest, followed by Beijing Airport. Both are in the global top 10.

In Africa-Middle East-Southern Asia, Middle Eastern airports dominate interregional traffic, with Dubai Airport leading. Hamad Airport in Qatar ranks third, and the second-largest UAE airport ranks fourth. Cairo Airport in Egypt is fifth. These airports are among the most connected interregional airports but not in the global top 10.

In South America, Guarulhos Airport in Brazil is the busiest, followed by El Dorado in Colombia. Ministro Pistarini in Argentina ranks third, and Rio G-T Jobim in Brazil is fourth. None of these airports are in the global top 20, but they are highly connected regarding routes.

World Transportation Network. The 25 busiest airports in the world air transportation network show that the North and Central America-Caribbean region controls a big part of the world's traffic. Indeed, 19 are in this area. Five are in the Europe-Russia-Central Asia-Central Asia region, and one is in the East and Southeast Asia-Oceania. Comparing the airports' strengths with their degree, one detects that European airports usually deserve more destinations worldwide, while North American airports have more flights.

In North and Central America-Caribbean, 19 airports rank in the world's top 25 for traffic, indicating regional dominance in global air transportation. However, in terms of routes, the European region holds sway. Six of the top 25 airports are inter-regional, while all are in the top 25 for regional traffic. This shows a focus on regional traffic within this region, mirroring the route dominance. London Heathrow Airport is third globally, primarily due to interregional traffic, while Charles de Gaulle, Frankfurt am Main, Amsterdam Schiphol, and Munich airports rank 6th, 7th, 12th, and 21st worldwide. They are all in the top 25 for regional and inter-regional traffic, with a stronger regional influence. Notably, London Heathrow is in the American-Caribbean component, with much denser traffic than airports in that region. A similar observation applies to Frankfurt am Main Airport, part of the Africa-Middle East-Southern Asia component. Beijing Airport is the sole East and Southeast Asia representative in the top 25 airports, ranking 18th globally for flight numbers. It holds the top position regionally and globally in terms of connectivity. Despite its extensive routes, Dubai Airport is the world's 45th busiest, leading in the Africa-Middle East-Southern Asia region but ranking 25th regionally and 19th inter-regionally. The most influential airport in South America, Guarulhos-Governador André F M Airport in São Paulo, Brazil, ranks 75th

globally in traffic, first regionally, and 24th inter-regionally. No airport in this region is among the top 25 most connected.

To summarize, North America leads the world air transportation network traffic. The component structure shows that most of this traffic is regional. Indeed, the large global component, which captures the inter-regional flights, exhibits numerous airports from different world areas essential to interregional traffic. In addition, the large local components display the influential regional airports hidden in the world air network.

3.2 RBO Analysis

The Ranked-Biased Overlap (RBO) [30] assesses the similarity of ranking lists, allowing for different weightings and evaluation depths. Its value ranges from 0 to 1, with higher values indicating greater similarity. We use RBO to compare the strength and degree centrality of large component airports [14], equalizing rank importance. Figure 3 displays RBO trends for top-ranked airports in large components and the global air transportation network, ranging from the top 5 to the top 45 with 5-unit steps.

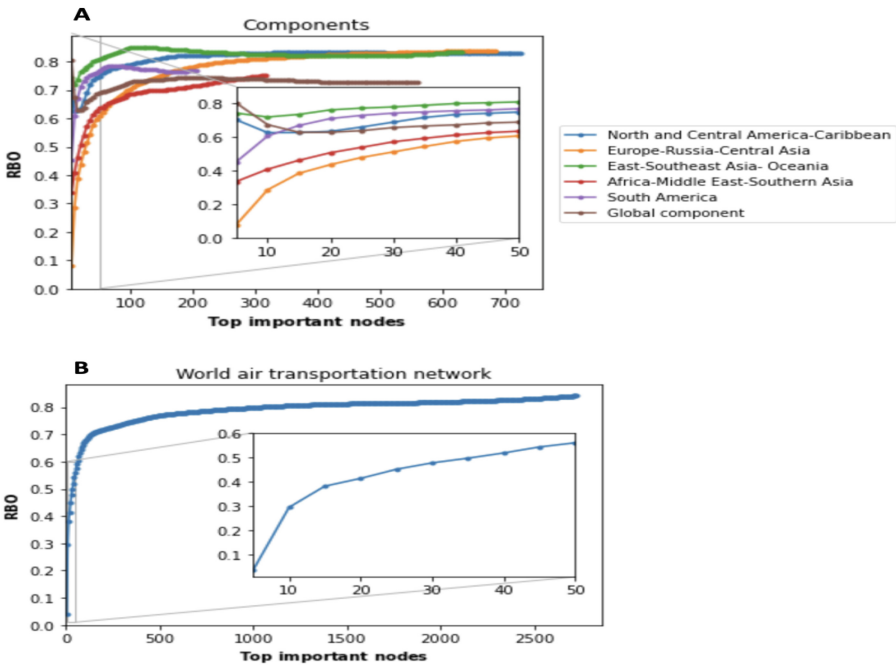


Fig. 3. The RBO of the large components (A) and the world air transportation network (B). The top 50 hubs of the unweighted (degree) and weighted (strength) are compared by step of 5 in the enlarged area.

Two curve categories emerge. In the first, RBO steadily rises with more airports, seen in Europe-Russia-Central Asia, Africa-Middle East-Southern Asia, East-Southeast Asia-Oceania, and South America components. These regions lack concentration on major hubs; their top 5 airports for degree and strength differ. In the second category, RBO initially decreases, then rises. This applies to North and Central America-Caribbean, East-Southeast Asia-Oceania, and the global component. The top 5 airports' degree and strength rankings are similar here. Differences emerge from the top 10 to 15, lowering RBO. Beyond that, rankings converge, increasing RBO. Analyzing the Top 45 airports, East-Southeast Asia-Oceania exhibits the closest alignment between traffic (strength) and hub size (degree). In contrast, Europe-Russia-Central Asia shows the most disparate ranking, with traffic concentrated in hubs, while airports with few connections handle significant traffic.

In the global air transportation network, RBO between top strength and degree airports steadily rises, revealing pronounced differences. The top 5 degree-focused airports cluster in Europe, whereas the top 5 strength-focused airports are in the United States. This contrast underscores distinct regional priorities: the USA emphasizes regional traffic, while Europe prioritizes serving numerous international destinations.

4 Discussion and Conclusion

In summary, this study provides a comprehensive analysis of the mesoscopic structure of the world air transportation network, shedding light on the diverse dynamics within and between regions. It underscores the critical roles played by specific hubs and highlights the varying regional and interregional characteristics.

Two distinct categories emerge in the analysis of local components' Max *s*-cores. The first category encompasses regions like Europe-Russia-Central Asia and East and Southeast Asia, where the Max *s*-core spans multiple countries. In the second pattern, which is found in regions with high traffic, only a few countries dominate. The Max *s*-core within local components comprises fewer airports than the Max *k*-core, indicating that a select few airports handle a substantial volume of flights and routes. In the global component, the Max *s*-core spans North and Central America-Caribbean, Europe-Russia-Central Asia, and East and Southeast Asia-Oceania regions, hubs for inter-regional flights. Unlike local components, the global Max *s*-core and Max *k*-core sizes are similar, emphasizing concentrated inter-regional traffic hubs. The global network's Max *s*-core predominantly lies in the USA. In contrast, Max *k*-core airports are in Europe. Thus, considering the global air network rather than the component structure can lead to a blurred understanding of the most essential airports.

The centrality analysis of the world air transportation network components reveals several key findings. The top five airports with the highest strength are typically situated in leading countries of their respective regions, although Europe-Russia-Central Asia shows a more balanced distribution. Interestingly, airports located far from their component's core can also rank among the top high-strength airports. High-degree airports, responsible for facilitating numerous flights, are prominent in North and Central America-Caribbean and East and Southeast Asia components. However, this pattern is less evident in other local components. Inter-regional airports with high strength are

spread across various countries, except for the USA, which dominates inter-regional traffic in North and Central America-Caribbean. These airports differ from the leading regional airports, indicating specialization in regional or inter-regional connectivity. Europe plays a significant role in inter-regional traffic.

This comparative analysis illustrates the essential contribution of the component structure representation for uncovering the regional and inter-regional similarities and differences of the world air transportation network.

References

- Orman, K., Labatut, V., Cherifi, H.: An empirical study of the relation between community structure and transitivity. In: Menezes, R., Evsukoff, A., González, M. (eds.) *Complex Networks*, pp. 99–110. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-30287-9_11
- Orman, G.K., Labatut, V., Cherifi, H.: Towards realistic artificial benchmark for community detection algorithms evaluation. *Int. J. Web Based Communities* **9**(3), 349–370 (2013)
- Lasfar, A., Mouline, S., Aboutajdine, D., Cherifi, H.: Content-based retrieval in fractal coded image databases. In: *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 1, pp. 1031–1034. IEEE (2000)
- Demirkesen, C., Cherifi, H.: A comparison of multiclass SVM methods for real world natural scenes. In: Blanc-Talon, J., Bourennane, S., Philips, W., Popescu, D., Scheunders, P. (eds.) *ACIVS 2008. LNCS*, vol. 5259, pp. 752–763. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88458-3_68
- Hamidi, M., Chetouani, A., El Haziti, M., El Hassouni, M., Cherifi, H.: Blind robust 3d mesh watermarking based on mesh saliency and wavelet transform for copyright protection. *Information* **10**(2), 67 (2019)
- Gupta, N., Singh, A., Cherifi, H.: Community-based immunization strategies for epidemic control. In: *2015 7th International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–6. IEEE (2015)
- Rajeh, S., Savonnet, M., Leclercq, E., Cherifi, H.: Interplay between hierarchy and centrality in complex networks. *IEEE Access* **8**, 129717–129742 (2020)
- Ghalmane, Z., Cherifi, C., Cherifi, H., El Hassouni, M.: Extracting backbones in weighted modular complex networks. *Sci. Rep.* **10**(1), 15539 (2020)
- Cherifi, H., Palla, G., Szymanski, B.K., Lu, X.: On community structure in complex networks: challenges and opportunities. *Appl. Network Sci.* **4**(1), 1–35 (2019)
- Prabhakar, N., Anbarasi, L.J.: Exploration of the global air transport network using social network analysis. *Soc. Network Anal. Mining* **11**, 1–12 (2021)
- Rocha, L.E.C.: Dynamics of air transport networks: a review from a complex systems perspective. *Chin. J. Aeronaut.* **30**(2), 469–478 (2017)
- Guimera, R., Mossa, S., Turttschi, A., Amaral, L.A.N.: The worldwide air transportation network: anomalous centrality, community structure, and cities' global roles. *Proc. Natl. Acad. Sci.* **102**(22), 7794–7799 (2005)
- Lordan, O., Sallan, J.M.: Core and critical cities of global region airport networks. *Physica A Stat. Mech. Appl.* **513**, 724–733 (2019)
- Diop, I.M., Cherifi, C., Diallo, C., Cherifi, H.: Revealing the component structure of the world air transportation network. *Appl. Network Sci.* **6**(1), 1–50 (2021)
- Md Murad Hossain and Sameer Alam: A complex network approach towards modeling and analysis of the Australian airport network. *J. Air Transp. Manag.* **60**, 1–9 (2017)

16. Cheung, T.K.Y., Wong, C.W.H., Zhang, A.: The evolution of aviation network: global airport connectivity index 2006–2016. *Transp. Res. Part E: Logist. Transp. Rev.* **133**, 101826 (2020)
17. Barrat, A., Barthelemy, M., Pastor-Satorras, R., Vespignani, A.: The architecture of complex weighted networks. *Proc. Natl. Acad. Sci.* **101**(11), 3747–3752 (2004)
18. Flightaware. <https://flightaware.com/>
19. Alves, L.G.A., Aleta, A., Rodrigues, F.A., Moreno, Y., Amaral, L.A.N.: Centrality anomalies in complex networks as a result of model over-simplification. *New J. Phys.* **22**(1), 013043 (2020)
20. Kong, Y.-X., Shi, G.-Y., Rui-Jie, W., Zhang, Y.-C.: k-core: theories and applications. *Phys. Rep.* **832**, 1–32 (2019)
21. Lăce, L., Chen, D., Ren, X.-L., Zhang, Q.-M., Zhang, Y.-C., Zhou, T.: Vital nodes identification in complex networks. *Physics Reports* **650**, 1–63 (2016)
22. Das, K., Samanta, S., Pal, M.: Study on centrality measures in social networks: a survey. *Soc. Netw. Anal. Min.* **8**(1), 13 (2018)
23. Ibnoulouafi, A., El Haziti, M., Cherifi, H.: M-centrality: identifying key nodes based on global position and local degree variation. *J. Stat. Mech: Theory Exp.* **2018**(7), 073407 (2018)
24. Rajeh, S., Savonnet, M., Leclercq, E., Cherifi, H.: Characterizing the interactions between classical and community-aware centrality measures in complex networks. *Sci. Rep.* **11**(1), 1–15 (2021)
25. Rajeh, S., Savonnet, M., Leclercq, E., Cherifi, H.: Comparative evaluation of community-aware centrality measures. *Qual. Quan.* **57**(2), 1273–1302 (2023)
26. Ghalmane, Z., El Hassouni, M., Cherifi, C., Cherifi, H.: Centrality in modular networks. *EPJ Data Sci.* **8**(1), 15 (2019)
27. Kumar, M., Singh, A., Cherifi, H.: An efficient immunization strategy using overlapping nodes and its neighborhoods. In: *Companion Proceedings of the The Web Conference 2018*, pp. 1269–1275 (2018)
28. Chakraborty, D., Singh, A., Cherifi, H.: Immunization strategies based on the overlapping nodes in networks with community structure. In: Nguyen, H.T.T., Snares, V. (eds.) *CSoNet 2016*. LNCS, vol. 9795, pp. 62–73. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42345-6_6
29. Blöcker, C., Nieves, J.C., Rosvall, M.: Map equation centrality: community-aware centrality based on the map equation. *Appl. Network Sci.* **7**(1), 56 (2022)
30. Webber, W., Moffat, A., Zobel, J.: A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst. (TOIS)* **28**(4), 1–38 (2010)

Diffusion and Epidemics



New Seeding Strategies for the Influence Maximization Problem

Seok-Hee Hong^{1(✉)}, Juan Pablo Bonilla Ataidés¹, Rowena Kok¹,
Amyra Meidiana¹, and Kunsoo Park²

¹ University of Sydney, Camperdown, Australia

seokhee.hong@sydney.edu.au,

{jbon1036,rkok7745,amei2916}@uni.sydney.edu.au

² Seoul National University, Seoul, Korea

kpark@snu.ac.kr

Abstract. We present two new seeding strategies for the Influence Maximization Problem for Viral Marketing, based on graph connectivity and spectral graph theory. Specifically, the first approach CVSP uses the *cut vertices* and the *separation pairs* as the starting seeds. The second approach ER uses the vertex ranking based on the *effective resistance* values of the incident edges. CVSP and ER are efficient, and can be implemented in linear and near linear time, respectively.

Experiments using the Independent Cascade diffusion model with real-world data sets show that our new seeding strategies perform significantly better than the existing methods, such as centrality measures, k -core and the state-of-the-art IMM, in particular for the scale-free networks with globally sparse, locally dense clusters with small diameters, in the final influence spread. Moreover, visual analysis enables more refined comparison between the methods, demonstrating that our methods have more globally wide influence spread pattern than other methods with locally dense influence spread pattern.

Keywords: Influence Maximization · Viral Marketing · Network Visualization

1 Introduction

In today's fast developing technology and increasing information availability, network models are growing increasingly complex. Coupled with the substantial growth of social media in recent years, businesses have realized the potential of analyzing these social networks, particularly with the notion of Viral Marketing.

Through Viral Marketing, a business' marketing message is transmitted by word-of-mouth exchanges in an exponentially growing manner [10]. The Viral Marketing Strategy can be extremely effective because it is crucially based on trust within individuals' social network [28]. In this way, influence and product adoption can spread from a few key *seed* influencers to a large portion of users in the network. Analyzing the spread of influence of Viral Marketing may also

This work is supported by ARC grant DP190103301.

have applications in other areas, such as epidemiology, since Viral Marketing is considered analogous to the transmission of infectious disease in an epidemic.

Unfortunately, choosing these initial seed users such that influence is maximized, known as the Influence Maximization Problem (IMP), is shown to be NP-hard [11]. It is difficult to efficiently choose the optimal seed set, especially with a limited budget [5]. Nevertheless, a number of heuristics and algorithms are presented in the literature, see Sect. 2.3 for details.

In this paper, we introduce two new seeding strategies for the IMP problem. The first strategy CVSP is based on the *connectivity* of the graph, i.e., the decomposition from one-connected graphs into biconnected components, and then from biconnected graphs into triconnected components [6, 7]. Specifically, we use *cut vertices* as well as *separation pairs* as the starting seeds. The second strategy ER is based on the *spectral* graph theory. Specifically, we use a vertex ranking defined by the *effective resistance* values [24, 25].

Extensive comparison experiments using real-world data sets demonstrate that our new seeding strategies outperform the existing methods [3, 8, 14, 17, 20], such as various centrality measures, k -core index as well as the state-of-the-art IMM [26], in particular for the scale-free networks with globally sparse, locally dense clusters with short diameter.

Specifically, we experiment using the most widely used diffusion model, Independent Cascade (IC), and compare the final influence spread rate with existing seeding strategies. For scale-free graphs with small diameters, ER and CVSP outperform, with the final influence spread rate of 71.9% and 67.6% respectively, compared to IMM with 50.5%. For biological networks with long diameter, IMM performs the best with the final influence spread of 52.1%, followed by ER and CVSP with 35.1% and 29.5%, respectively.

Since cut vertices and separation pairs can be computed in linear time [6, 7], and the effective resistance values can be computed in near linear time [24], our new seeding strategies CVSP and ER are not only effective, but also efficient, matching the near linear time complexity of IMM [26]. Furthermore, visual analysis allows more sophisticated comparison between the methods, showing that our methods have more global influence spread pattern than other methods with local influence spread pattern.

2 Related Work

2.1 Influence Maximization Problem

The Influence Maximization Problem (IMP) [13] involves finding a set of seed vertices in a network such that the eventual spread of influence under a stochastic diffusion model is maximized. The problem is shown as NP-hard [11] and has specific applications in Viral Marketing [4, 19, 22].

Roughly speaking, the spread of influence in a network under a diffusion model is modelled as follows: (i) A set of k seed vertices are chosen and *activated*; (ii) A diffusion model, which imitates the spread of influence, repeatedly runs

through a new set of activated vertices at each diffusion step, until the spread converges (i.e., no new vertices are activated). Let $I(S)$ denote the number of activated vertices when the diffusion process finishes using the initial set of activated vertices, the *seed set* S . Let $G = (V, E, \omega)$ be a graph with a vertex set V and the edge set E , where $\omega(u, v)$ represents the weight of an edge (u, v) . Let M be a stochastic diffusion model with a positive integer k , representing the budget. Formally, the Influence Maximization Problem asks to find a seed set S^* such that $S^* \subseteq V \wedge |S^*| = k$ maximizes $\mathbb{E}[I(S^*)]$.

2.2 Diffusion Models

The most widespread influence propagation diffusion models are the Independent Cascade model and the Linear Threshold model, which are used as a general framework for IMP [12].

The *Independent Cascade (IC)* model generalizes the Susceptible-Infected-Recovered (SIR) model, one of the most ubiquitous diffusion models [23]. In these models, all vertices are initially susceptible (able to be infected) except for the infected vertices (the initial seed vertices), which only have a single chance to infect (influence) their neighbors.

At each time step k , an infected vertex u will independently infect each of its susceptible neighbors v with probability β in the SIR model, and $\beta_{u,v}$ in the IC model. At step $k + 1$, every infected vertex from the previous step enters the recovered state. This diffusion process ends once there are no more infected vertices and thus no more possible activation. The spreading influence of the initial seed set is defined as the number of resulting recovered vertices [20]. The IC model is often used to simulate influence propagation and it models, for example, how independent recommendations and reviews may induce a purchase [15].

In contrast, the *Linear Threshold (LT)* model describes a process in which all vertices that were active in the last time step remain active, and thus each vertex's tendency to become active increases monotonically [13]. This is in line with the notion that an individual's perceived value of a product increases as more people in their network adopt the product.

2.3 Seeding Strategies for the IMP

Kempe et al. [13] introduce the two diffusion models IC and LT, and then show that under these two models, the IMP problem is NP-Hard. They propose a greedy algorithm with a guarantee that the final influence spread is within $(1 - \frac{1}{e} - \epsilon)$, where ϵ is a tunable parameter. However, the greedy algorithm has poor scalability. Leskovec et al. [16] present the CELF algorithm, using the submodularity properties of the influence spread function for both IC and LT models to terminate the greedy algorithm early, with a 700 times speed up with the same performance guarantee.

Since then, a plethora of methods have been developed to solve the IMP, either heuristics or with performance guarantees for the sake of runtime. For example, IMM by Tang et al. [26] is one of the most well-known algorithms

considered as the state-of-the-art, based on the Reverse Influence Sampling (RIS) framework by Borgs et al. [2], which runs in near linear time, i.e., $O((k+l)(n+m) \log n/\epsilon^2)$, where l and ϵ are tunable parameters, $n = |V|$, $m = |E|$, $k = |S|$. IMM can return a $(1 - \frac{1}{e} - \epsilon)$ approximation with at least $1 - \frac{1}{n^t}$ probability.

Since the IMP is NP-hard, many heuristic algorithms are presented, where the typical method is to rank all vertices according to a certain measure and select the k highest ranking vertices as the initial seed set. With the goal of identifying the most important and connected vertices, natural measures to utilize are *centrality* measures and k -core index from social network analysis [27].

The *Degree centrality* of a vertex, defined as the number of adjacent vertices, is a simple but intuitive index used to identify a vertex's influence. Despite its simplicity and low computation cost, Liu et al. [17] report that when the spreading rate is small, Degree centrality outperforms Eigenvector centrality and the k -core index. However, Chen et al. [3] argue that Degree centrality has low accuracy and propose the *Semi-Local Centrality* (SLC) measure, which considers both the nearest and the next nearest neighbors of a vertex.

The *Closeness centrality* of a vertex is defined based on the sum of the shortest paths from the vertex to all the other vertices, to reflect how efficiently a vertex exchanges information with others [20]. Closeness centrality has been shown to outperform Degree and Betweenness centrality in terms of spreading influence [3]. However, computing the Closeness centrality requires significantly more time (i.e., $O(n^3)$ time [9]) than other measures, such as Degree centrality and k -core, its practical application can be limited.

The *Betweenness centrality* of a vertex is defined based on the number of the shortest paths that pass through the vertex, representing its potential power in controlling the information flow in a network. In contrast to Chen et al. [3], Lu et al. find that Betweenness centrality outperforms Closeness centrality in terms of the ratio of final recovered (infected) vertices [20]. Unfortunately, the high computational complexity of Betweenness centrality (i.e., $O(n^3)$ for general graphs; for sparse graphs, $O(nm)$ time for unweighted graphs and $O(nm + n^2 \log n)$ for weighted graphs [9]) means that for large networks, its scalability is infeasible.

The k -core of a graph G is the maximal subgraph where all vertices have degree at least k . Specifically, the k -core decomposition assigns an integer (coreness) k_s to each vertex, representing its location in the network based on its k -shell layer, where a vertex with larger coreness indicates that it is located in a more central, *core* position. The efficiency (i.e., linear time [1]) of k -core decomposition allows it to be applied to large-scale networks. Moreover, Kitsak et al. [14] find that the most efficient spreaders in the SIR model are those in the core of the network as identified by the k -core analysis, rather than vertices that are highly connected. However, Liu, et al. [18] find that due to the low granularity of k -core decomposition, higher k -core vertices often link locally to other vertices within the shell and, as such, are not good spreaders.

3 New Seeding Strategies

3.1 CVSP: Connectivity-Based Seeding Strategy

Our first strategy, CVSP (Cut Vertex Separation Pair), is based on the *connectivity* of the graph, specifically the decomposition from one-connected graphs into biconnected components, and then from biconnected graphs into triconnected components.

A *cut vertex* of a connected graph G is a vertex whose removal from G increases the components in G , i.e., disconnects [6]. Similarly, a *separation pair* of a biconnected graph G is a pair of vertices whose removal from G increases the components in G , i.e., disconnects [7]. Note that the cut vertices and separation pairs can be computed in linear time [6, 7].

Cut vertices and separation pairs are structurally important in terms of graph connectivity and topology, since the removal of such vertices disconnect the remaining graph, meaning that they disable communication and increase the vulnerability. In particular, cut vertices have been shown to be important actors in social network analysis, such as brokers, often with high betweenness centrality [27], indicating their potential use as seed vertices for the IMP problem.

To compare the performance of CVSP to other seeding strategies, such as various centrality measures, with resource restrictions, such as the budget which are conventionally modelled by the number of starting seed vertices k , we need to compute a vertex ranking by sorting the cut vertices and the separation pairs based on their degree. Specifically, CVSP can be described as follows:

1. Compute cut vertices of a one-connected graph G and sort them in the decreasing order of their degree to form the set CV . Ties are broken randomly to avoid introducing bias.
2. For each biconnected component B_i of G , compute the separation pairs to form the set SP . Sort the vertices in SP based on their degree, as in Step 1.
3. Compute the ordered *connectivity set* C by attaching SP at the end of CV . Select the top k vertices from C as the starting seed set.

3.2 ER: Spectral Seeding Strategy

Our second strategy, ER (Effective Resistance), is based on the spectral graph theory, which is concerned with the eigenvalues and eigenvectors of matrices associated with graphs. The *spectrum* of a graph is the list of eigenvalues of its Laplacian matrix L , which is defined using the Degree matrix D and Adjacency matrix A as $L = D - A$. The spectrum of a graph is closely related to a number of structural properties of graphs, such as the connectivity.

For example, a spectral sparsifier is a subgraph whose Laplacian quadratic form is approximately the same as the original on all real vector inputs. Spielman and Teng [25] proved that every n -vertex graph has a spectral approximation with $O(n \log n)$ edges. Specifically, *spectral sparsification* is a stochastic sampling method, where edges are sampled based on the *effective resistance* values, which is related to the commute distances in the graph.

Modelling a graph as an electrical network, the effective resistance of an edge is defined as the voltage drop across the edge and its value is equivalent to the probability of the edge to be included in a random spanning tree of the graph [24]. Note that the effective resistance values of edges in a graph can be computed in near linear time [24].

Since effective resistance values are closely related to the *commute distance* (i.e., the expected length of a random walk between two vertices), as well as *structural roles (positions)* of actors based on how they are embedded in a social network [27], these indicate potential use as seed vertices for the IMP problem. Specifically, ER can be described as follows:

1. Compute the effective resistance value $ER(v)$ of each vertex v as the sum of the effective resistance value $ER(e)$ of each incident edge e of v : i.e. $ER(v) = \sum_{e \in I(v)} ER(e)$, where $I(v)$ is the set of incident edges of v .
2. Compute a vertex ranking R in a decreasing order of $ER(v)$. Select the top k vertices from the vertex ranking R as the initial seed set.

4 Comparison Experiments

4.1 Experiment Design, Implementation and Data Sets

We compare our two new seeding strategies CVSP and ER against the following popular seeding strategies, described in Sect. 2.3: BC (Betweenness Centrality), CC (Closeness Centrality), DC (Degree Centrality), KC (k -core) and IMM.

We implement CVSP and ER, and diffusion simulations in Python. For the IC model, we use an implementation from [3], and initialize $\beta_{u,v}$ values to 0.5. We use NetworkX to compute centrality measures and k -core, and OGDF to compute the cut vertices, separation pairs and graph layouts. For IMM, we use implementation from [26] with the recommended setting of the tunable parameter $\epsilon = 0.5$.

We experiment with two types of real-world datasets: (1) *scale-free networks*: globally sparse, locally dense clusters with the Power law degree distribution and *short* diameters; (2) *biological networks* (RNA sequence graphs): globally sparse (i.e. path or almost tree structure), locally dense clusters with *long* diameters, called the *GION graphs* [21]. For the details of the data sets, see Table 1.

For each graph, we run the Independent Cascade (IC) diffusion model with each seeding strategy five times and take the average, due to the non-deterministic nature of the model. We set the budget (i.e., the number of starting seed vertices) as $k = |CV| + |SP|$.

4.2 Final Influence Spreading Rate Comparison

Table 1 and Fig. 1 show the *final influence spreading rate* (i.e., percentage of the activated vertices over the total number of vertices after IC converges). Clearly, ER performs the best on the scale-free graphs, followed by BC and CVSP, and DC performs on par with CVSP. Interestingly, IMM performs the worst for

Table 1. The average final influence spread rate (i.e., percentage of the activated vertices over the total number of vertices in the graph), where the number of starting seed vertices $k = |CV| + |SP|$, den means the density, and dia means the diameter of a graph. ER performs the best for scale-free graphs, followed by BC and CVSP; IMM performs the best for GION graphs, followed by ER and CVSP.

Dataset	BC	CC	DC	KC	CVSP	ER	IMM	$ V $	$ E $	den	dia	k
soc_h	58.8	53.2	54.8	51.2	53	60	33.8	2000	16097	8.05	10	326
yeastppi	83.8	74.4	81.6	73.8	83.6	85.6	59.4	2224	7049	3.17	11	674
offlights	81.2	65.4	75.6	68.2	76.4	81.4	44.2	2905	15645	5.39	14	709
facebook	49.2	47	45.6	9.8	46.8	49.2	52.6	4039	88234	21.8	8	109
p2p-Gnutella05	80.4	72.8	77.6	73.2	78.4	83.2	62.4	8842	31837	3.60	9	2636
Avg. Scale-free	70.68	62.56	67.04	55.24	67.64	71.88	50.48					
gion2	31	17.4	26.4	20.6	32.6	39.4	53.2	1159	6424	5.54	58	183
gion5	37.4	23.2	23.8	20.2	38.4	46.8	57.8	1748	13957	7.98	63	270
gion6	29.8	18.4	26.8	17.4	30.8	34	51.6	1785	20459	11.5	41	198
gion7	13.8	8.4	13.8	10.2	20.2	21.2	46.2	3010	41757	13.9	77	161
gion8	28.6	12.4	23.4	16.8	25.4	34	51.8	4924	52502	10.7	121	491
Avg. GION	28.12	15.96	22.84	17.04	29.48	35.08	52.12					

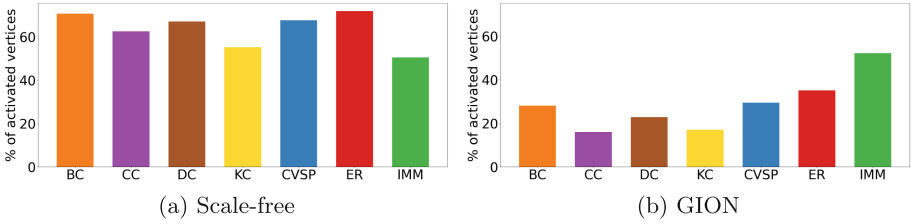


Fig. 1. Average final influence spread rate (i.e., percentage of activated vertices): ER performs the best for scale-free graphs; IMM performs the best for GION graphs.

scale-free graphs. On the other hand, IMM performs the best on the GION graphs, followed by ER and CVSP. Overall, experiments demonstrate excellent performance by ER and CVSP, in particular for scale-free graphs. Note that the time complexity of CVSP and ER is much faster than BC.

Figure 2 shows how fast the diffusion is carried out for different seeding strategies, and where the rate of diffusion peaks, by showing the number of newly activated vertices at each diffusion step. In terms of the influence rate, we observe that most of the diffusion occurs in the first few steps across all seeding strategies. The highest influence rate occurs at the second (or the third) diffusion step, and then drops quickly in particular for scale-free networks. Therefore, what determines which method will activate the most vertices after convergence is which method activates the most vertices in the first few diffusion steps. As such, the role of the starting seed vertices is critical.

Figure 3 shows the final influence rate after convergence, where the number of starting seed vertices k is varied. Clearly, the relative performances of seeding

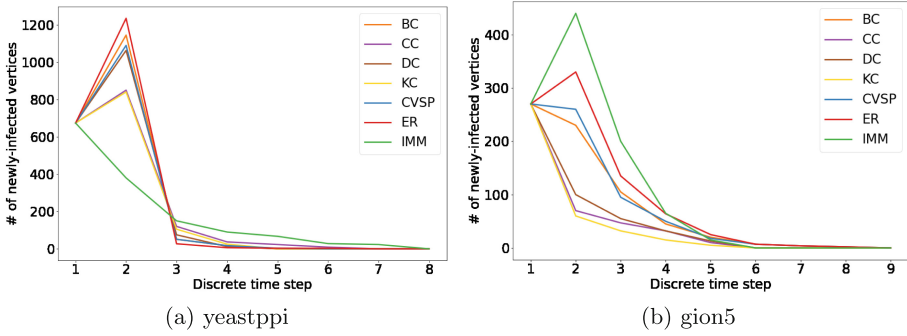


Fig. 2. Dynamics of diffusion: the number of newly activated vertices at each diffusion step. For all strategies, most of the diffusion occurs in the second step.

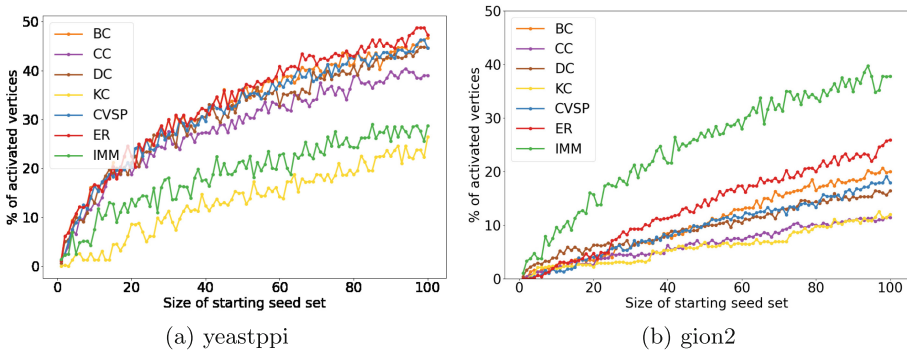


Fig. 3. Final influence rate after convergence, where the number of starting seed vertices k is varied. In general, the relative performances of seeding strategies do not change, based on the size of k .

strategies do not change, based on the size of k . We observe that varying the number of starting seed vertices k has minimal effect on the results observed in Table 1. Roughly speaking, when a particular seeding strategy is strong relative to the rest, it remains strong for all k values, and when two seeding strategies are comparable, they remain comparable for all k .

4.3 Visual Analysis and Comparison

Visual analysis enables more refined comparison between the methods other than the efficiency and effectiveness. For example, in terms of efficiency, CVSP, DC and KC all run in linear time. For scale-free graphs, Table 1 shows that both CVSP and DC perform on par in terms of effectiveness (i.e., influence spreading rate). However, visual comparison in Fig. 4 reveals that the seed vertices by CVSP have a more global influence compared to the seed vertices by DC or KC with relatively local influence.

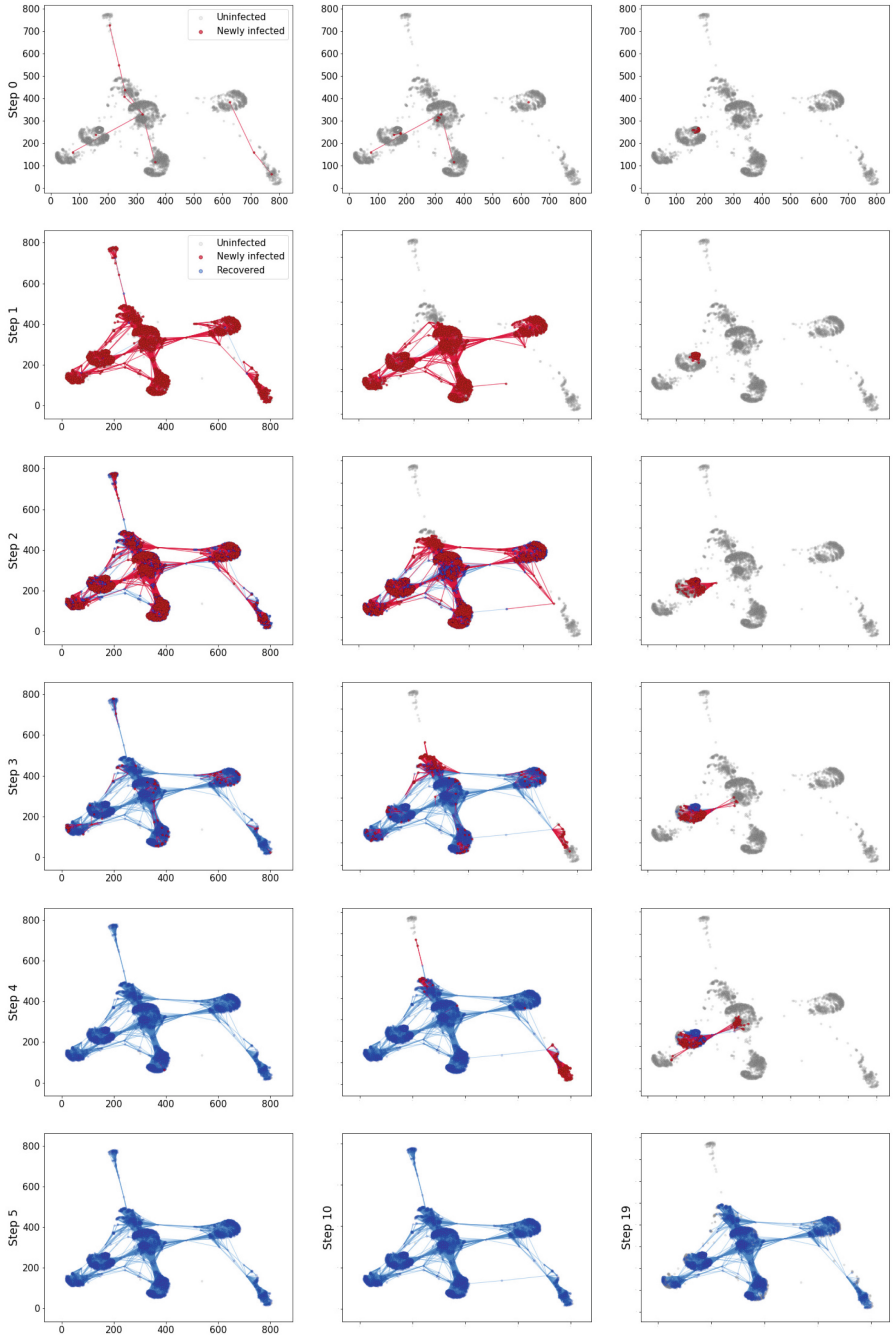


Fig. 4. Visual analysis and comparison of the influence spread by CVSP (left), DC (middle) and KC (right) on the facebook network. Note that the final diffusion step in the last row is 5 (CVSP), 10 (DC) and 19 (KC), respectively. Red (resp., Blue) means newly (resp., previously) activated vertices/edges, and Grey means not yet activated.

Specifically, we can observe that in the first two steps, CVSP produces more widely spread influenced vertices (in red), compared to more concentrated, sustained local influence spread by DC and KC, suggesting that their seed vertices are highly locally linked, leading to redundancy. Moreover, the final influence (in blue recovered vertices) of CVSP, DC and KC are achieved in 5, 10 and 19 diffusion steps respectively (in the last row), implying that the initial seed vertices by CVSP are more effective influencers than the seeds by DC and KC.

4.4 Summary and Recommendation

In summary, extensive comparison experiments demonstrate that our new seeding strategies CVSP and ER are not only efficient with linear and near-linear time complexity, but also effective in terms of the influence spreading rate, as shown in Table 1. Overall, ER and CVSP outperform IMM for scale-free graphs with globally sparse, locally dense clusters with ultra small diameters. On the other hand, IMM performs the best for GION graphs, followed by ER and CVSP.

We also conducted experiments with the LT diffusion model, and obtained similar results. Based on our extensive empirical experiments, we conclude with the following recommendation and guidelines:

- For scale-free graphs with globally sparse, locally dense clusters and short diameters, we recommend ER for both effectiveness and efficiency.
- For graphs with globally sparse, locally dense clusters and long diameters, we recommend IMM for both effectiveness and efficiency.
- When efficiency is the most important consideration, we recommend CVSP over DC or KC, for global influence spread pattern, based on visual analysis and comparison.

References

1. Batagelj, V., Mrvar, A., Zaveršnik, M.: Partitioning approach to visualization of large graphs. In: GD 1999, pp. 90–97 (1999)
2. Borgs, C., Brautbar, M., Chayes, J., Lucier, B.: Maximizing social influence in nearly optimal time. In: SODA 2014, pp. 946–957 (2014)
3. Chen, D., Lü, L., Shang, M.S., Zhang, Y.C., Zhou, T.: Identifying influential nodes in complex networks. *Phys. A* **391**(4), 1777–1787 (2012)
4. Domingos, P., Richardson, M.: Mining the network value of customers. In: SIGKDD 2001, pp. 57–66 (2001)
5. Gui-sheng, Y., Ji-jie, W., Hong-bin, D., Jia, L.: Intelligent viral marketing algorithm over online social network. In: ICNDC 2011, pp. 319–323 (2011)
6. Hopcroft, J., Tarjan, R.: Algorithm 447: efficient algorithms for graph manipulation. *Commun. ACM* **16**(6), 372–378 (1973)
7. Hopcroft, J.E., Tarjan, R.E.: Dividing a graph into triconnected components. *SIAM J. Comput.* **2**(3), 135–158 (1973)
8. Iyer, S., Killingback, T., Sundaram, B., Wang, Z.: Attack robustness and centrality of complex networks. *PLoS ONE* **8**(4), e59,613 (2013)

9. Jacob, R., Koschützki, D., Lehmann, K.A., Peeters, L., Tenfelde-Podehl, D.: Algorithms for centrality indices. In: Brandes, U., Erlebach, T. (eds.) *Network Analysis*. LNCS, vol. 3418, pp. 62–82. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31955-9_4
10. Kaplan, A.M., Haenlein, M.: Two hearts in three-quarter time: how to waltz the social media/viral marketing dance. *Bus. Horiz.* **54**(3), 253–263 (2011)
11. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: *SIGKDD 2003*, pp. 137–146 (2003)
12. Kempe, D., Kleinberg, J., Tardos, É.: Influential nodes in a diffusion model for social networks. In: *ICALP 2005*, pp. 1127–1138 (2005)
13. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. *Theory Comput.* **11**(4), 105–147 (2015)
14. Kitsak, M., et al.: Identification of influential spreaders in complex networks. *Nat. Phys.* **6**(11), 888–893 (2010)
15. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. *ACM Trans. Web (TWEB)* **1**(1), 5–es (2007)
16. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: *SIGKDD 2007, KDD 2007*, pp. 420–429 (2007)
17. Liu, J.G., Lin, J.H., Guo, Q., Zhou, T.: Locating influential nodes via dynamics-sensitive centrality. *Sci. Rep.* **6**(1), 1–8 (2016)
18. Liu, Y., Tang, M., Zhou, T., Do, Y.: Core-like groups result in invalidation of identifying super-spreader by k-shell decomposition. *Sci. Rep.* **5**(1), 1–8 (2015)
19. Long, C., Wong, R.C.W.: Viral marketing for dedicated customers. *Inf. Syst.* **46**, 1–23 (2014)
20. Lü, L., Chen, D., Ren, X.L., Zhang, Q.M., Zhang, Y.C., Zhou, T.: Vital nodes identification in complex networks. *Phys. Rep.* **650**, 1–63 (2016)
21. Marnier, M.R., Smith, R.T., Thomas, B.H., Klein, K., Eades, P., Hong, S.H.: Gion: interactively untangling large graphs on wall-sized displays. In: *GD 2014*, pp. 113–124 (2014)
22. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: *SIGKDD 2002*, pp. 61–70 (2002)
23. Shakarian, P., Bhatnagar, A., Aleali, A., Shaabani, E., Guo, R.: *Diffusion in Social Networks*. SCS, Springer, Cham (2015). <https://doi.org/10.1007/978-3-319-23105-1>
24. Spielman, D.A., Srivastava, N.: Graph sparsification by effective resistances. *SIAM J. Comput.* **40**(6), 1913–1926 (2011)
25. Spielman, D.A., Teng, S.H.: Spectral sparsification of graphs. *SIAM J. Comput.* **40**(4), 981–1025 (2011)
26. Tang, Y., Shi, Y., Xiao, X.: Influence maximization in near-linear time: a martingale approach. In: *SIGMOD 2015*, pp. 1539–1554 (2015)
27. Wasserman, S., Faust, K., et al.: *Social network analysis: methods and applications* (1994)
28. Zhu, T., Wang, B., Wu, B., Zhu, C.: Maximizing the spread of influence ranking in social networks. *Inf. Sci.* **278**, 535–544 (2014)



Effects of Homophily in Epidemic Processes

Richard J. La^{1,2}(✉)

¹ Applied and Computational Mathematics Division, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA
richard.la@nist.gov

² University of Maryland, College Park, MD 20742, USA

Abstract. We examine how homophily widely observed in networks affects epidemic processes on networks, which are often used to model the spread of information and diseases in social networks and of malware and viruses in information systems. We propose a new framework for modeling the effects of homophily with the help of multi-type branching processes. Using this framework we derive conditions that tell us how homophily influences the likelihood of suffering an epidemic. Finally, we present numerical studies to verify our findings.

Keywords: epidemic processes · homophily · multi-type branching processes

1 Introduction

Studies revealed that both engineered and natural networks exhibit several interesting properties, such as the small world phenomenon, high transitivity (or clustering), degree correlations to name a few [1, 6–8]. These properties influence how diseases, information or viruses spread over networks, which is often modeled using epidemic processes on graphs or networks. For this reason, researchers examined how they affect the dynamics of epidemic processes on networks, e.g., [3, 4].

Another property that has been observed widely in social networks is *homophily* [5]. It refers to the observation that people tend to be friends with others who share common beliefs or have similar background, and is shown to have significant impact on our decisions. We are interested in investigating how homophily influences the spread of infectious diseases in a population. For example, a close circle of friends may share similar beliefs, including their opinions about vaccines and the risks from an infectious disease. As a result, when one member of the circle is antivaccine or does not believe in (the effectiveness of) vaccines, other members may be less likely to get vaccinated (than the general public) and be more likely to contract the disease when exposed. We note that, although we study the effects of homophily in the context of a disease epidemic, our framework and results apply to other applications, including information or virus propagation in information networks and systems.

Many studies demonstrated homophily due to different relationships in social networks [5]. However, our understanding of its impact on epidemic processes is still limited. We consider a population consisting of two types of individuals, e.g., vaccinated vs. unvaccinated. First, we propose a new framework for estimating the probability of experiencing an epidemic (PoEP) in a large population, starting with a single infected

individual at the beginning. Second, making use of this framework, we carry out an analysis and derive a set of conditions that tell us how homophily changes the PoEPs (see Theorem 1). Finally, we conduct numerical studies to validate our findings.

The rest of the paper is organized as follows: We introduce the basic setup in Sect. 2. Section 3 first describes the model and approach, including multi-type branching processes, and then presents the main results. Numerical results are provided in Sect. 4. We conclude in Sect. 5.

2 Basic Setup

Consider a population consisting of N individuals, where N is large, and let $\mathcal{V} := \{1, \dots, N\}$ be the set of individuals in the population. We model the existence of a relation between two distinct individuals i and j in \mathcal{V} using a bidirectional link between them, which is denoted by (i, j) . Thus, we represent the relations among the individuals using an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of undirected links indicating the existing relations. We call \mathcal{G} the relation graph (RG).

We assume that each individual is assumed to be one of two types, either type 1 or 2, and let $\mathcal{T} = \{1, 2\}$ be the set of types. In the earlier example, type 1 (resp. type 2) may correspond to *vaccinated* individuals (resp. *unvaccinated* individuals). Let $p \in (0, 1)$ denote the fraction of type 1 individuals in the population. For each $i \in \mathcal{T}$, define λ_i to be the probability with which a type i individual will contract the disease when exposed to an infected neighbor.

Suppose that the degree distribution in the population is given by $\mathbf{v} = (v_k : k \in \mathcal{D})$, where $\mathcal{D} := \{1, \dots, d_{\max}\}$ and $v_k, k \in \mathcal{D}$, is the fraction of population with degree k . Define d_{avg} to be the mean degree given by $d_{\text{avg}} = \sum_{k \in \mathcal{D}} k \cdot v_k$. We denote the degree distribution of a (randomly chosen) neighbor by $\mathbf{w} = (w_k : k \in \mathcal{D})$, where $w_k = k \cdot v_k / d_{\text{avg}}$. Here, we assume that the degree distribution does not depend on the type of individuals as we are interested in isolating the effects of homophily on epidemic processes.

In the absence of homophily, the probability that a neighbor of an individual is of type 1 would be p . In order to study the effects of homophily on epidemic processes, we introduce different probabilities based on the individual's type: for a type i individual ($i \in \mathcal{T}$), each of its neighbors is a type 1 individual with probability q_i , independently of each other. When $q_1 = q_2 = p$, it indicates no bias introduced due to homophily. For each $i \in \mathcal{T}$, let $\theta_1^i := q_i \lambda_1$, $\theta_2^i := (1 - q_i) \lambda_2$, and $\theta_0^i := 1 - \theta_1^i - \theta_2^i$. Obviously, when $q_1 = q_2 = p$, we have $\theta_\ell^1 = \theta_\ell^2, \ell \in \{0, 1, 2\}$.

Clearly, when q_1 is fixed, q_2 cannot be an independent variable since the total number of outgoing edges is fixed to be $N \cdot d_{\text{avg}}$ and that of type 1 individuals (resp. type 2 individuals) is $N \cdot p \cdot d_{\text{avg}}$ (resp. $N(1 - p)d_{\text{avg}}$). In order to compute q_2 as a function of q_1 , we can solve the following equation using the total numbers of outgoing and incoming edges of type 1 individuals in the population:

$$\begin{aligned} \text{total \# of outgoing edges} &= N \cdot p \cdot d_{\text{avg}} \\ &= N \cdot p \cdot d_{\text{avg}} \cdot q_1 + N(1 - p)d_{\text{avg}} \cdot q_2 = \text{total \# of incoming edges} \end{aligned}$$

By solving for q_2 , we obtain $q_2 = \frac{p(1-q_1)}{1-p} = \alpha \cdot (1 - q_1)$ with $\alpha = \frac{p}{1-p}$.

2.1 Probability of Epidemics

Suppose that the patient zero infected with a disease at the beginning is a type i individual. We denote by μ_i the probability that the disease will affect only a negligible fraction of the population and not lead to a widespread epidemic, starting with the infected type i individual. Similarly, let ξ_i be the probability that a type i individual, when infected by a neighbor, will not cause a cascade of infections via other neighbors. We use μ and ξ to denote (μ_1, μ_2) and (ξ_1, ξ_2) , respectively.

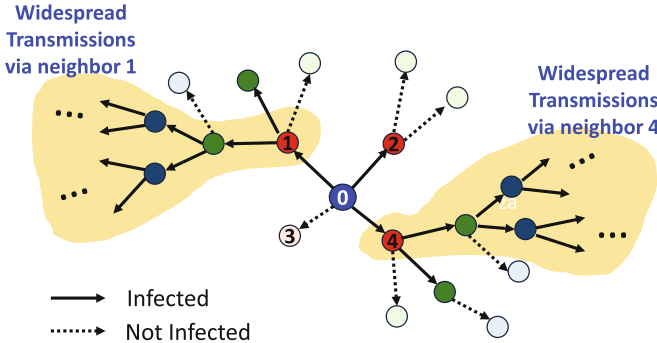


Fig. 1. Example scenario used to illustrate μ and ξ .

We illustrate these probabilities using the example shown in Fig. 1. Here, individual 0 (shown as a solid blue circle) is the patient zero who contracts the disease first and has 4 neighbors. Neighbors 1, 2, and 4 (shown as solid red circles) become infected by individual 0. But, neighbor 2 does not lead to widespread transmissions of the disease via other neighbors, whereas neighbors 1 and 4 cause cascades of infections through their neighbors (the sets of individuals infected through neighbors 1 and 4 are indicated by yellow areas). In this example, $1 - \mu_{i_0}$, where i_0 is the type of individual 0, is the probability that individual 0, following the contraction of disease, will cause an epidemic through its 4 neighbors, and the probability that an infected neighbor, say individual 1, will lead to an epidemic after getting infected by individual 0 is given by $1 - \xi_{i_1}$ with i_1 being its type.

Under suitable independence assumptions, the probability μ_i can be computed by conditioning on the degree of the patient zero as follows:

$$\mu_i(\xi, q_1) = \sum_{d=1}^{d_{\max}} v_d \left(1 - \underbrace{(q_i \lambda_1 (1 - \xi_1) + (1 - q_i) \lambda_2 (1 - \xi_2))}_{=\phi_i(\xi, q_1)} \right)^d, \quad (1)$$

where $\phi_i(\xi, q_1)$ is the probability that a randomly selected neighbor of the infected individual under consideration will trigger cascading infections. Equation (1) implicitly assumes that the clustering coefficient of the network is small so that we could treat the events of different neighbors causing cascading infections as independent events. For a study on the impact of clustering on epidemic processes, we refer the reader to [4].

3 Model and Main Results

3.1 Multi-type Branching Processes

It is clear from (1) that the probability μ_i is shaped by ξ . For this reason, we first focus on computing ξ . Although we assume only two types of individuals in \mathcal{T} , for the purpose of computing ξ using multi-type branching processes (MTBPs) explained below, we find it convenient to introduce two additional types: a type 3 individual (resp. a type 4 individual) is a type 1 individual (resp. a type 2 individual) who was infected by its neighbor. Hence, ξ_1 (resp. ξ_2) can be viewed as the probability that a single type 3 (resp. type 4) individual will not cause cascading infections through its other neighbors excluding the one that infected it. For each $i \in \mathcal{T}^+ := \{3, 4\}$, let $q_i = q_{i-2}$, $\lambda_i = \lambda_{i-2}$, $\theta_\ell^i = \theta_\ell^{i-2}$, $\ell \in \{0, 1, 2\}$.

In order to estimate ξ , we will make use of MTBPs: an MTBP starts with a single node of either type 3 or 4 at the beginning (generation 0), where a type i node ($i \in \mathcal{T}^+$) represents an infected type i individual described above. Starting with the single node in generation 0, each node in generation n , $n \in \mathbb{Z}_+ := \{0, 1, 2, \dots\}$, produces a set of nodes of types 3 and 4 in generation $n + 1$, which are called its *children*, according to some specified distribution (discussed below). These children in generation $n + 1$ represent the neighbors that are infected by the node in generation n .

Let P_n , $n \in \mathbb{Z}_+$, be the total number of nodes in generation n (with $P_0 = 1$). For each pair (n, l) , $n \in \mathbb{Z}_+$ and $l \in \{1, \dots, P_n\}$, let $C^{(n,l)} = (C_3^{(n,l)}, C_4^{(n,l)})$ be a random vector, where $C_i^{(n,l)}$, $i \in \mathcal{T}^+$, is the number of type i children produced by the l -th node in generation n . Clearly, for $n \in \mathbb{Z}_+$, we have $P_{n+1} = \sum_{l=1}^{P_n} (\sum_{i=3}^4 C_i^{(n,l)})$. The number and types of children produced by each node in an MTBP may depend on the node's type, but are assumed to be independent of those of other nodes. In other words, conditional on the types of P_n nodes in generation n , $C^{(n,l)}$, $l = 1, \dots, P_n$, are conditionally independent (and also independent of those of other generations). The probability of extinction (PoE) of the MTBP is defined to be the probability that there is some $n' \in \mathbb{Z}_+$ such that $P_{n'} = 0$ (with the convention $P_n = 0$ for all $n > n'$), i.e., the disease propagation stops after a finite number of hops. It is clear that the PoE may depend on the type of the single node in generation 0.

We approximate the probability ξ_i , $i \in \mathcal{T}$, using the PoE of an MTBP, starting with a single node of type $i + 2$ at the beginning. This is a reasonable assumption in a large population, in which a widespread epidemic that affects a non-negligible fraction of population must propagate over many hops via community transmissions, and successful prevention of an epidemic requires stopping the spread of a disease before community transmissions take hold.

Recall that an infected individual can have two different types of children, types 3 and 4. For fixed q_1 (hence fixed q_2 as well), the children distribution of a type i individual ($i \in \mathcal{T}^+$) can be computed by conditioning on the degree of the individual: for fixed $\mathbf{o} = (o_3, o_4) \in \mathbb{Z}_+^2$, the probability that an infected type i individual ($i \in \mathcal{T}^+$) will produce o_3 type 3 children and o_4 type 4 children is equal to

$$C_i(\mathbf{o}, q_1) = \sum_{d=o_3+o_4}^{d_{\max}-1} \left(w_{d+1} \frac{d!}{o_3!o_4!(d-o_3-o_4)!} (\theta_1^i)^{o_3} (\theta_2^i)^{o_4} (\theta_0^i)^{d-o_3-o_4} \right). \quad (2)$$

As expected, when $q_1 = q_2 = p$, C_1 and C_2 are identical.

For each $i \in \mathcal{T}^+$, with a little abuse of notation, let $\mathbb{E}[C_i] = (\mathbb{E}[C_{i,j}] : j \in \mathcal{T}^+)$ be a 1×2 row vector, whose j -th element is the expected number of type j children from an infected type i individual. Define $\mathbf{M} = [M_{i,j} : i, j \in \mathcal{T}^+]$ to be a 2×2 matrix, whose i -th row is $\mathbb{E}[C_i]$, i.e., $M_{i,j} = \mathbb{E}[C_{i,j}]$. Let $\rho(\mathbf{M})$ denote the spectral radius of \mathbf{M} . It is well known [2] that $\boldsymbol{\xi} = \mathbf{1} := (1, 1)$ if either (i) $\rho(\mathbf{M}) < 1$ or (ii) $\rho(\mathbf{M}) = 1$ and there is at least one type in \mathcal{T}^+ for which the probability that it produces exactly one child is not equal to one. Similarly, if $\rho(\mathbf{M}) > 1$, then $\boldsymbol{\xi} < \mathbf{1}$ and there is strictly positive probability that the spread of infection continues forever in the MTBP (in an infinite population), suggesting that there could be a widespread epidemic in a finite but large population.

3.2 Generating Functions

Unfortunately, there is no closed-form expression for $\boldsymbol{\xi}$. However, it can be computed as a fixed point of generating functions described in this subsection. For each type $i \in \mathcal{T}^+$, the corresponding generating function $f_i : [0, 1]^2 \rightarrow [0, 1]^2$ is given by¹

$$f_i(\mathbf{s}, q_1) = \sum_{\mathbf{o} \in \mathbb{Z}_+^2} C_i(\mathbf{o}, q_1) s_1^{o_3} s_2^{o_4}, \quad \mathbf{s} \in [0, 1]^2, q_1 \in (0, 1), \tag{3}$$

where C_i is the children distribution of a type i individual in (2). Substituting (2) in (3) and interchanging the order of summations, we obtain

$$\begin{aligned} & f_i(\mathbf{s}, q_1) \\ &= \sum_{\mathbf{o} \in \mathbb{Z}_+^2} \left[\sum_{d=o_3+o_4}^{d_{\max}-1} \left(w_{d+1} \frac{d!}{o_3!o_4!(d-o_3-o_4)!} (\theta_1^i)^{o_3} (\theta_2^i)^{o_4} (\theta_0^i)^{d-o_3-o_4} \right) s_1^{o_3} s_2^{o_4} \right] \\ &= \sum_{d \in \mathbb{Z}_+} w_{d+1} \left(\sum_{\{\mathbf{o} \in \mathbb{Z}_+^2 \mid o_3+o_4 \leq d\}} \frac{d!}{o_3!o_4!(d-o_3-o_4)!} (s_1\theta_1^i)^{o_3} (s_2\theta_2^i)^{o_4} (\theta_0^i)^{d-o_3-o_4} \right) \\ &= \sum_{d \in \mathbb{Z}_+} w_{d+1} \underbrace{(s_1\theta_1^i + s_2\theta_2^i + \theta_0^i)^d}_{=\psi_i(\mathbf{s}, q_1)}, \tag{4} \end{aligned}$$

where the last equality follows from the well-known equality

$$\sum_{\{\mathbf{n} \in \mathbb{Z}_+^m \mid \sum_{k=1}^m n_k = \tilde{N}\}} \frac{\tilde{N}!}{n_1!n_2! \cdots n_m!} \prod_{k=1}^m (x_k)^{n_k} = \left(\sum_{k=1}^m x_k \right)^{\tilde{N}}.$$

Fix $q_1 \in (0, 1)$. When $\rho(\mathbf{M}) < 1$, a unique fixed point of $\mathbf{f}(\mathbf{s}, q_1) := (f_3(\mathbf{s}, q_1), f_4(\mathbf{s}, q_1))$ is $\mathbf{1}$ and $\xi_1 = \xi_2 = 1$. On the other hand, if $\rho(\mathbf{M}) > 1$, there are two fixed points of $\mathbf{f}(\mathbf{s}, q_1)$ in $[0, 1]^2$ – one fixed point is $\mathbf{1}$ and $\boldsymbol{\xi}$ is the other fixed point in $[0, 1]^2$.

¹ For our discussion, we limit the domain and co-domain of the generating functions to $[0, 1]^2$.

In other words, $\xi = f(\xi, q_1) < 1$. When $\rho(\mathbf{M}) > 1$ for fixed $q_1 \in (0, 1)$, we denote the fixed point $\xi < 1$ by $\xi(q_1)$ to make the dependence on q_1 explicit.

We are interested in understanding how the fixed point $\xi(q_1)$ changes as q_1 varies, especially around p . In other words, we wish to examine how homophily observed in many social networks affects the likelihood of suffering an epidemic, starting with a random infection of an individual. One may suspect that when type 1 individuals are more likely to be neighbors with each other than with type 2 individuals, it would be more difficult to infect their neighbors and consequently $\mu_1(\xi(q_1), q_1)$ would increase with q_1 . However, this intuition is not always correct and the following theorem suggests that the answer to the question is far more complicated and the effects of homophily depend on many other factors. Define $\mu(\xi(q_1), q_1) := p \cdot \mu_1(\xi(q_1), q_1) + (1 - p)\mu_2(\xi(q_1), q_1)$.

Theorem 1. Fix $p \in (0, 1)$, and suppose $\xi(p) = (\xi_1(p), \xi_2(p)) \in (0, 1)^2$. Let

$$A := \sum_{d=1}^{d_{\max}-1} (w_{d+1}d(1 - (1 - \xi)(\lambda_2 - p(\lambda_2 - \lambda_1)))^{d-1}),$$

where $\xi := \xi_1(p) = \xi_2(p)$, and define

$$K := \frac{A(\lambda_2 - \lambda_1)}{1 - A(p\lambda_1 + (1-p)\lambda_2)}, \tag{5a}$$

$$\Gamma_1 := 1 - Kp \quad \text{and} \quad \Gamma_2 := 1 + K(1 - p). \tag{5b}$$

- i.** If $\Gamma_1 > 0$ (resp. $\Gamma_1 < 0$), then there exists $\epsilon_1 > 0$ such that, for all $p - \epsilon_1 < q_1 \leq \tilde{q}_1 < p + \epsilon_1$, $\mu_1(\xi(q_1), q_1) \leq \mu_1(\xi(\tilde{q}_1), \tilde{q}_1)$ (resp. $\mu_1(\xi(q_1), q_1) \geq \mu_1(\xi(\tilde{q}_1), \tilde{q}_1)$);
- ii.** If $\Gamma_2 > 0$ (resp. $\Gamma_2 < 0$), then there exists $\epsilon_2 > 0$ such that, for all $p - \epsilon_2 < q_1 \leq \tilde{q}_1 < p + \epsilon_2$, $\mu_2(\xi(q_1), q_1) \geq \mu_2(\xi(\tilde{q}_1), \tilde{q}_1)$ (resp. $\mu_2(\xi(q_1), q_1) \leq \mu_2(\xi(\tilde{q}_1), \tilde{q}_1)$); and
- iii.** If $K > 0$ (resp. $K < 0$), then there exists $\epsilon > 0$ such that, for all $p - \epsilon < q_1 \leq \tilde{q}_1 < p + \epsilon$, $\mu(\xi(q_1), q_1) \geq \mu(\xi(\tilde{q}_1), \tilde{q}_1)$ (resp. $\mu(\xi(q_1), q_1) \leq \mu(\xi(\tilde{q}_1), \tilde{q}_1)$).

Remark 1. Before we proceed, we remark on some observations and a relation between Γ_1 and Γ_2 . First, the theorem tells us that whether or not type i individuals benefit from homophily in that they become less likely to cause an epidemic depends on the sign of Γ_i defined in (5b). Moreover, loosely speaking, Γ_1 and Γ_2 tend to move in the opposite directions. This is consistent with earlier intuition that μ_1 and μ_2 should move in opposite directions with increasing homophily. However, our findings in the theorem tell us that this intuition is only partially correct.

Second, suppose $\Gamma_2 < 0$. From the definitions of Γ_1 and Γ_2 in (5b), this implies $K < -(1 - p)^{-1} < 0$ and $\Gamma_1 > 0$. Similarly, when $\Gamma_1 < 0$, we have $K > p^{-1} > 0$ and $\Gamma_2 > 0$. Therefore, it is not possible for both Γ_1 and Γ_2 to be negative. This in turn means that it is impossible for μ_1 to decrease and μ_2 to increase simultaneously due to homophily (locally around p), which is expected.

4 Numerical Studies

Our main analytical result in Theorem 1 tells us how weak homophily affects the likelihood of experiencing an epidemic, starting with a single infected individual, when

$q_1 \approx p$. However, it does not tell us how the likelihood would change in the presence of strong homophily. In this section, we study how the parameter q_1 , which determines the level of homophily present in the network, affects the PoEP with the help of numerical studies. As we will see shortly, it turns out that its effect is not straightforward and depends on other parameters and, ultimately, the current PoEP of each type.

For our study, we consider a scale-free network in which the node degrees follow a power law with parameter 1.5: let $\mathbf{v} := (v_d : d \in \{d_{\min}, \dots, d_{\max}\})$ be the degree distribution, where $v_d \propto d^{-1.5}$, $d = d_{\min}, \dots, d_{\max}$. The minimum degree d_{\min} and the maximum degree d_{\max} are set to 2 and 25, respectively. Given the parameters, the mean degree is 6.28.

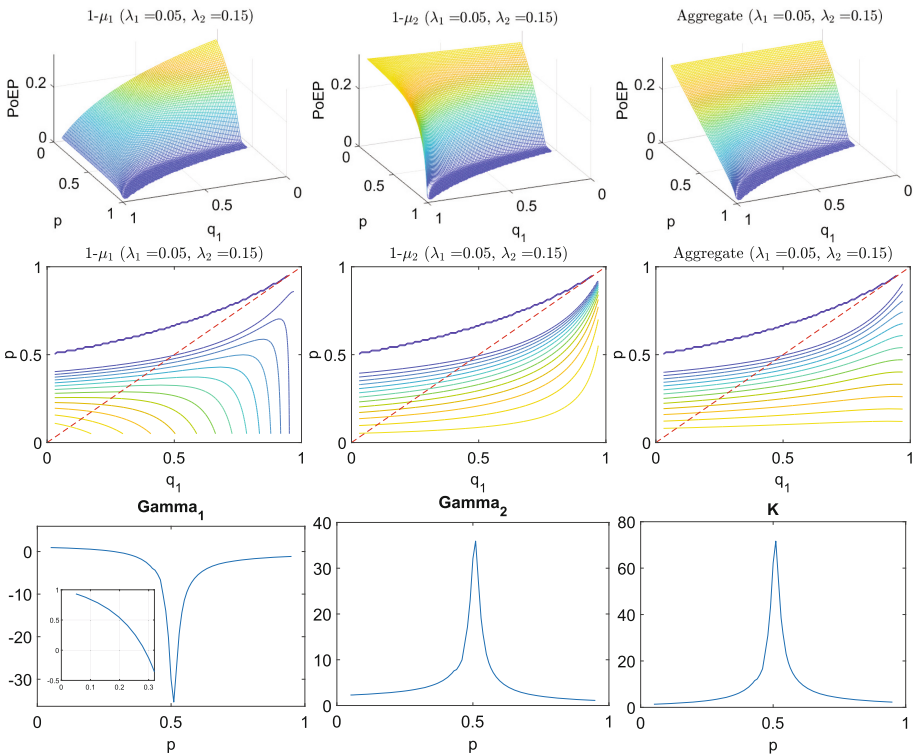


Fig. 2. Small to moderate PoEP case. (i) Plots of PoEPs (top row), (ii) contours of PoEPs (middle row), and (iii) plots of Γ_1 , Γ_2 , and K (bottom row).

• *Small to moderate PoEP regime* – In the first case, we consider a scenario in which the PoEP is small to moderate when $p = q_1 = q_2$. For this scenario, we choose $\lambda_1 = 0.05$ and $\lambda_2 = 0.15$. Figure 2 shows (i) $\bar{\mu}_1 := 1 - \mu_1$, $\bar{\mu}_2 := 1 - \mu_2$, and the overall PoEP given by $\bar{\mu} := p \cdot \bar{\mu}_1 + (1 - p)\bar{\mu}_2$ (top row), (ii) the contour plots of the PoEPs shown in the first row (second row), and (iii) Γ_1 , Γ_2 and K defined in (5) as a function of p (bottom row).

There are several observations that we can make from the plots. First, it is clear from the plots that, in the absence of homophily, i.e., $q_1 = p$, we have $\bar{\mu}_1 = \bar{\mu}_2$. This is expected because the right-hand side of (1) is the same for both μ_1 and μ_2 when $p = q_1 = q_2$. Also, the PoEP $\bar{\mu}$ decreases with the fraction of vaccinated individuals in the population, namely p , as they are less vulnerable to infection (from $\lambda_1 < \lambda_2$).

Second, when p is small ($p < 0.28$), weak homophily with $q_1 \approx p$ tends to reduce $\bar{\mu}_1$ for type 1 individuals. This can be seen in the contour plot for $\bar{\mu}_1$ and the plot of Γ_1 , which is positive when $p < 0.28$.² However, for larger $p > 0.28$, it has the opposite effect on $\bar{\mu}_1$ as Γ_1 dips below zero. On the other hand, for all $p \in (0, 1)$, $\bar{\mu}_2$ increases with q_1 . Similarly, weighted average $\bar{\mu}$ tends to increase with q_1 around p for all $p \in (0, 1)$, and this can be verified by the plot of K , which is strictly positive for all $p \in (0, 1)$. This trend continues for q_1 outside a local neighborhood around p . The observation suggests that, at least in some cases, homophily may have adverse effects

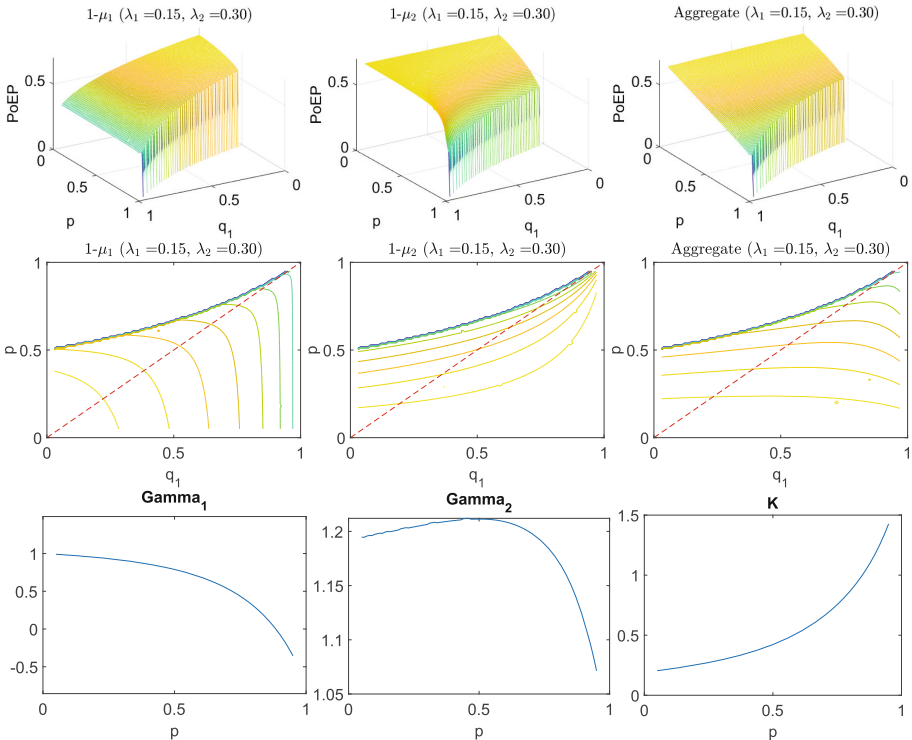


Fig. 3. Moderate PoEP case. (i) Plots of PoEPs (top row), (ii) contours of PoEPs (middle row), and (iii) plots of Γ_1 , Γ_2 , and K (bottom row).

² Recall that each line in the contour plots represents a set of pairs (p, q_1) where the value of PoEP is the same. When the slope of the contour line at the dotted red line (45° line) is positive (resp. negative), it indicates that the PoEP increases (resp. decreases) with homophily. Also, if homophily does not affect PoEP, the contour lines would be flat.

and facilitate the spread of an epidemic, thereby increasing the likelihood of suffering an epidemic. Also, it tells us that, in order to accurately predict and avoid underestimating the likelihood of suffering an epidemic from an infectious disease, it is important to consider homophily in the network.

• *Higher PoEP regime* – In the second case, we examine a scenario in which the PoEP is much larger in the absence of homophily. To this end, we increase the infection probabilities to $\lambda_1 = 0.15$ and $\lambda_2 = 0.3$. Figure 3 shows (i) $\bar{\mu}_1$, $\bar{\mu}_2$ and $\bar{\mu}$, (ii) their contour plots, and (iii) Γ_1 , Γ_2 and K . It is clear from the plot that $\bar{\mu}_1$ decreases with q_1 over for a wide range of p , while $\bar{\mu}_2$ still increases with q_1 . This is confirmed by the plots of Γ_i , $i = 1, 2$. In addition, $\bar{\mu}$ also increases with q_1 in the neighborhood around p as verified by the plot of K . However, this increase in PoEP with q_1 is not as pronounced as in the first case (shown in Fig. 2).

5 Conclusion

We studied the influence of homophily observed in many social networks. With the help of MTBPs, we proposed a new framework for modeling the effects of homophily and approximating the probability of experiencing an epidemic starting with a single infected individual. Our analytical results tell us how weak homophily changes the likelihood of suffering an epidemic. Finally, we validated our analytical findings using numerical studies.

A Proof of Theorem 1

From (1), we have, for each $i = 1, 2$,

$$\frac{\partial \mu_i(\boldsymbol{\xi}(q_1), q_1)}{\partial q_1} = \sum_{d=1}^{d_{\max}} v_d d (1 - \phi_i(\boldsymbol{\xi}(q_1), q_1))^{d-1} \left(- \frac{\partial \phi_i(\boldsymbol{\xi}(q_1), q_1)}{\partial q_1} \right). \tag{6}$$

It is clear from (6) that the sign of $\frac{\partial \mu_i(\boldsymbol{\xi}(q_1), q_1)}{\partial q_1}$, $i = 1, 2$, depends on that of $\frac{\partial \phi_i(\boldsymbol{\xi}(q_1), q_1)}{\partial q_1}$. Specifically, their signs are the opposite. For this reason, we focus on computing the partial derivatives $\frac{\partial \phi_i(\boldsymbol{\xi}(q_1), q_1)}{\partial q_1}$, $i = 1, 2$. First, from the definition of ϕ_i , we have

$$\frac{\partial \phi_1(\boldsymbol{\xi}(q_1), q_1)}{\partial q_1} = \lambda_1(1 - \xi_1) - \lambda_2(1 - \xi_2) - q_1 \lambda_1 \frac{\partial \xi_1(q_1)}{\partial q_1} - (1 - q_1) \lambda_2 \frac{\partial \xi_2(q_1)}{\partial q_1} \tag{7}$$

In order to evaluate the partial derivatives $\partial \xi_i / \partial q_1$, $i = 1, 2$, we make use of the implicit function theorem (IFT) [9]: define $h_i(\mathbf{s}, q_1) := f_{i+2}(\mathbf{s}, q_1) - s_i$, $i = 1, 2$. Recall that $\boldsymbol{\xi}(p) \in (0, 1)^2$ satisfies $h_i(\boldsymbol{\xi}(p), p) = 0$, $i = 1, 2$. Thus, the IFT tells us the following: (i) if the Jacobian matrix $\mathbf{M} = [\partial h_i(\boldsymbol{\xi}, p) / \partial \xi_j : i, j = 1, 2]$ is invertible, we can find an open interval $(p - \epsilon, p + \epsilon)$ such that $\boldsymbol{\xi}(q_1)$ is *continuously differentiable* over the interval, and (ii) the following equality holds:

$$\begin{bmatrix} \frac{\partial \xi_1(q_1)}{\partial q_1} \\ \frac{\partial \xi_2(q_1)}{\partial q_1} \end{bmatrix} = - \begin{bmatrix} \frac{\partial h_1(\boldsymbol{\xi}, q_1)}{\partial \xi_1} & \frac{\partial h_1(\boldsymbol{\xi}, q_1)}{\partial \xi_2} \\ \frac{\partial h_2(\boldsymbol{\xi}, q_1)}{\partial \xi_1} & \frac{\partial h_2(\boldsymbol{\xi}, q_1)}{\partial \xi_2} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial h_1(\boldsymbol{\xi}, q_1)}{\partial q_1} \\ \frac{\partial h_2(\boldsymbol{\xi}, q_1)}{\partial q_1} \end{bmatrix} \tag{8}$$

We now proceed to compute each partial derivative in \mathbf{M} .

$$\begin{aligned} \frac{\partial h_1(\xi, q_1)}{\partial \xi_1} &= A_1 \theta_1^1 - 1 = A_1 q_1 \lambda_1 - 1 \\ \frac{\partial h_1(\xi, q_1)}{\partial \xi_2} &= A_1 \theta_2^1 = A_1 (1 - q_1) \lambda_2 \\ \frac{\partial h_2(\xi, q_1)}{\partial \xi_1} &= A_2 \theta_1^2 = A_2 \frac{p}{1-p} (1 - q_1) \lambda_1 \\ \frac{\partial h_2(\xi, q_1)}{\partial \xi_2} &= A_2 \theta_2^2 - 1 = A_2 \frac{1 - 2p + pq_1}{1-p} \lambda_2 - 1 \end{aligned}$$

where

$$A_i = \sum_{d=1}^{d_{\max}-1} w_{d+1} d (1 - (1 - \xi_1) \theta_1^i - (1 - \xi_2) \theta_2^i)^{d-1} > 0, \quad i = 1, 2.$$

Note that when $q_1 = p$, we have $\xi_1 = \xi_2 = \xi$ and $\theta_k^1 = \theta_k^2, k = 1, 2$. Evaluating these partial derivatives at $q_1 = p$, we obtain

$$\mathbf{M} = \begin{bmatrix} Ap\lambda_1 - 1 & A(1-p)\lambda_2 \\ Ap\lambda_1 & A(1-p)\lambda_2 - 1 \end{bmatrix},$$

where we use A to denote $A_1 = A_2 = \sum_{d=1}^{d_{\max}-1} (w_{d+1} d (1 - (1 - \xi)(\lambda_2 - p(\lambda_2 - \lambda_1)))^{d-1}) > 0$. The inverse of matrix \mathbf{M} is given by

$$\mathbf{M}^{-1} = \frac{1}{\det(\mathbf{M})} \begin{bmatrix} A(1-p)\lambda_2 - 1 & -A(1-p)\lambda_2 \\ -Ap\lambda_1 & Ap\lambda_1 - 1 \end{bmatrix}, \tag{9}$$

where $\det(\mathbf{M}) = 1 - A(p\lambda_1 + (1-p)\lambda_2) \leq 1 - A\lambda_1 < 0$. Following similar steps and evaluating at $q_1 = p$,

$$\begin{aligned} \left. \frac{\partial h_1(\xi, q_1)}{\partial q_1} \right|_{q_1=p} &= A \frac{\partial}{\partial q_1} (\xi_1 q_1 \lambda_1 + \xi_2 (1 - q_1) \lambda_2 + 1 - q_1 \lambda_1 - (1 - q_1) \lambda_2) \Big|_{q_1=p} \\ &= A(\lambda_2 - \lambda_1)(1 - \xi) \end{aligned} \tag{10}$$

and

$$\begin{aligned} \left. \frac{\partial h_2(\xi, q_1)}{\partial q_1} \right|_{q_1=p} &= A \frac{\partial}{\partial q_1} (\xi_1 \frac{p}{1-p} (1 - q_1) \lambda_1 + \xi_2 \frac{1 - 2p + pq_1}{1-p} \lambda_2 \\ &\quad + 1 - \frac{p}{1-p} (1 - q_1) \lambda_1 - \frac{1 - 2p + pq_1}{1-p} \lambda_2) \Big|_{q_1=p} \\ &= -\frac{Ap}{1-p} (\lambda_2 - \lambda_1)(1 - \xi). \end{aligned} \tag{11}$$

Substituting (9), (10) and (11) in (8),

$$\begin{aligned} \begin{bmatrix} \frac{\partial \xi_1(q_1)}{\partial q_1} \\ \frac{\partial \xi_2(q_1)}{\partial q_1} \end{bmatrix} \Big|_{q_1=p} &= -\frac{A(\lambda_2 - \lambda_1)(1 - \xi)}{\det(\mathbf{M})} \begin{bmatrix} A(1-p)\lambda_2 - 1 & -A(1-p)\lambda_2 \\ -Ap\lambda_1 & Ap\lambda_1 - 1 \end{bmatrix} \begin{bmatrix} 1 \\ -\frac{p}{1-p} \end{bmatrix} \\ &= B \begin{bmatrix} A(1-p)\lambda_2 - 1 + Ap\lambda_2 \\ -Ap\lambda_1 - Ap\lambda_1 \frac{p}{1-p} + \frac{p}{1-p} \end{bmatrix} = B \begin{bmatrix} A\lambda_2 - 1 \\ -\frac{Ap(1-p)\lambda_1 - Ap^2\lambda_1 + p}{1-p} \end{bmatrix} = B \begin{bmatrix} A\lambda_2 - 1 \\ -\frac{p(A\lambda_1 - 1)}{1-p} \end{bmatrix} \end{aligned}$$

where $B = -A(\lambda_2 - \lambda_1)(1 - \xi)/\det(\mathbf{M})$.

Substituting these expressions in (7) and evaluating at $q_1 = p$,

$$\begin{aligned} &\frac{\partial \phi_1(\xi(q_1), q_1)}{\partial q_1} \Big|_{q_1=p} \\ &= \lambda_1(1 - \xi) - \lambda_2(1 - \xi) - p\lambda_1 B(A\lambda_2 - 1) - (1-p)\lambda_2 B\left(\frac{-p(A\lambda_1 - 1)}{1-p}\right) \\ &= \lambda_1(1 - \xi) - \lambda_2(1 - \xi) - p\lambda_1 B(A\lambda_2 - 1) + p\lambda_2 B(A\lambda_1 - 1) \\ &= -(\lambda_2 - \lambda_1)(1 - \xi + pB). \end{aligned} \tag{12}$$

From the definitions of B and the determinant of \mathbf{M} ,

$$1 - \xi + pB = (1 - \xi) \left(1 - \frac{Ap(\lambda_2 - \lambda_1)}{1 - A(p\lambda_1 + (1-p)\lambda_2)} \right) = (1 - \xi)\Gamma_1,$$

where Γ_1 is defined in (5b). Since $\lambda_2 > \lambda_1$ and $\xi \in (0, 1)$, when $\Gamma_1 > 0$, (12) is strictly negative and $\frac{\partial \phi_1(\xi(q_1), q_1)}{\partial q_1} \Big|_{q_1=p} < 0$. Similarly, if $\Gamma_1 < 0$, then (12) is strictly positive and we have $\frac{\partial \phi_1(\xi(q_1), q_1)}{\partial q_1} \Big|_{q_1=p} > 0$.

Following similar steps,

$$\begin{aligned} &\frac{\partial \phi_2(\xi(q_1), q_1)}{\partial q_1} \Big|_{q_1=p} \\ &= -\lambda_1(1 - \xi) \frac{p}{1-p} + \lambda_2(1 - \xi) \frac{p}{1-p} - p\lambda_1 \frac{\partial \xi_1}{\partial q_1} - (1-p)\lambda_2 \frac{\partial \xi_2}{\partial q_1} \\ &= -\frac{p\lambda_1(1 - \xi)}{1-p} + \frac{p\lambda_2(1 - \xi)}{1-p} - p\lambda_1 B(A\lambda_2 - 1) + p\lambda_2 B(A\lambda_1 - 1) \\ &= p(\lambda_2 - \lambda_1) \left(\frac{1 - \xi}{1-p} - B \right) = p(\lambda_2 - \lambda_1)(1 - \xi) \left(\frac{1}{1-p} + \frac{A(\lambda_2 - \lambda_1)}{\det(\mathbf{M})} \right) \\ &= \frac{p(\lambda_2 - \lambda_1)(1 - \xi)}{1-p} \left(1 + \frac{A(1-p)(\lambda_2 - \lambda_1)}{1 - A(p\lambda_1 + (1-p)\lambda_2)} \right) \\ &= \frac{p(\lambda_2 - \lambda_1)(1 - \xi)}{1-p} \Gamma_2, \end{aligned} \tag{13}$$

where Γ_2 is defined in (5b). When $\Gamma_2 > 0$, (13) is strictly positive and $\frac{\partial \phi_2(\xi(q_1), q_1)}{\partial q_1} \Big|_{q_1=p} > 0$. On the other hand, if $\Gamma_2 < 0$, (13) is strictly negative and $\frac{\partial \phi_2(\xi(q_1), q_1)}{\partial q_1} \Big|_{q_1=p} < 0$.

Finally, from the definition of $\mu(\xi(q_1), q_1)$,

$$\frac{\partial \mu(\xi(q_1), q_1)}{\partial q_1} = p \frac{\partial \mu_1(\xi(q_1), q_1)}{\partial q_1} + (1 - p) \frac{\partial \mu_2(\xi(q_1), q_1)}{\partial q_1}.$$

Substituting the expression in (6),

$$\begin{aligned} \frac{\partial \mu(\xi(q_1), q_1)}{\partial q_1} = & p \sum_{d=1}^{d_{\max}} v_d d (1 - \phi_1(\xi(q_1), q_1))^{d-1} \left(- \frac{\partial \phi_1(\xi(q_1), q_1)}{\partial q_1} \right) \\ & + (1 - p) \sum_{d=1}^{d_{\max}} v_d d (1 - \phi_2(\xi(q_1), q_1))^{d-1} \left(- \frac{\partial \phi_2(\xi(q_1), q_1)}{\partial q_1} \right). \end{aligned}$$

Evaluating at $q_1 = p$ and substituting the expressions in (12) and (13),

$$\begin{aligned} \left. \frac{\partial \mu(\xi(q_1), q_1)}{\partial q_1} \right|_{q_1=p} &= -p \Sigma \left. \frac{\partial \phi_1(\xi(q_1), q_1)}{\partial q_1} \right|_{q_1=p} - (1 - p) \Sigma \left. \frac{\partial \phi_2(\xi(q_1), q_1)}{\partial q_1} \right|_{q_1=p} \\ &= -\Sigma \left(p (- (\lambda_2 - \lambda_1)(1 - \xi) \Gamma_1) + (1 - p) \frac{p(\lambda_2 - \lambda_1)(1 - \xi)}{1 - p} \Gamma_2 \right) \\ &= -\Sigma (\lambda_2 - \lambda_1)(1 - \xi)(\Gamma_2 - \Gamma_1) = -\Sigma (\lambda_2 - \lambda_1)(1 - \xi)K, \end{aligned}$$





where $\Sigma := \sum_{d=1}^{d_{\max}} v_d d (1 - \phi_1(\xi(p), p))^{d-1} = \sum_{d=1}^{d_{\max}} v_d d (1 - \phi_2(\xi(p), p))^{d-1} = \sum_{d=1}^{d_{\max}} v_d d (1 - (1 - \xi)(p\lambda_1 + (1 - p)\lambda_2))^{d-1} > 0$.

References

1. Boguñá, M., Pastor-Satorras, R., Vespignani, A.: Epidemic spreading in complex networks with degree correlations. *Lect. Not. Phys.* **625**, 127–147 (2003)
2. Harris, T.E.: *The Theory of Branching Processes*. Springer-Verlag, Heidelberg (1963)
3. La, R.J.: Cascading failures in interdependent systems: impact of degree variability and dependence. *IEEE Trans. Network Sci. Eng.* **5**(2), 127–140 (2018)
4. La, R.J.: Influence of clustering on cascading failures in interdependent systems. *IEEE Trans. Network Sci. Eng.* **6**(3), 351–363 (2019)
5. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: homophily. *Ann. Rev. Sociol.* **27**, 415–444 (2001)
6. Newman, M.E.J.: The structure and function of complex networks. *SIAM Rev.* **45**(2), 167256 (2003)
7. Newman, M.E.J.: *Networks*. Oxford University Press, Oxford (2010)
8. Pastor-Satorras, R., Castellano, C., Van Mieghem, P., Vespignani, A.: Epidemic processes in complex networks. *Rev. Mod. Phys.* **87**, 925–979 (2015)
9. Rudin, W.: *Principles of Mathematical Analysis*. McGraw-Hill, New York (1976)



Human Papillomavirus Co-circulation on a Partially Vaccinated Partnership Network

Mélanie Bonneault^{1,2,3} , Maxime Flauder^{1,2},
Elisabeth Delarocque-Astagneau² , Anne C. M. Thiébaud³ ,
and Lulla Opatowski^{1,2} 

¹ Epidemiology and Modelling of Antibiotic Evasion Unit, Institut Pasteur, 75015 Paris, France
melanie.bonneault@gmail.com

² Université Paris-Saclay, UVSQ, Inserm, CESP, 78180 Montigny-Le-Bretonneux, France

³ Université Paris-Saclay, UVSQ, Inserm, CESP, 94807 Villejuif, France

Abstract. Human papillomaviruses (HPV) are among the most common sexually transmitted infections and a necessary cause of cervical cancer. In the context of vaccination against a sub-group of genotypes, better understanding the respective role of biological interactions between HPV genotypes and social interactions between humans is essential to anticipate what the vaccine impact could be at the population level. Here, we present a novel stochastic agent-based model formalizing the co-circulation on a human partnership network of multiple interacting genotypes, some of them being preventable by the vaccine (vaccine types) and others not. The model explicitly formalizes heterogeneity in sexual behaviors and allows exploration of distinct genotypic interaction mechanisms during intra-host co-infections. Using model simulations, we investigate infection dynamics after vaccine introduction in the population depending on assumptions about vaccine coverage and interactions between vaccine and non-vaccine genotypes.

Keywords: agent-based model · sexually transmitted infection · genotype interactions

1 Introduction

Mathematical models are useful tools to understand the dynamics of infectious disease spread in human populations and to predict the impact of control strategies such as screening or immunization programs. Classically, individuals are grouped indiscriminately in compartments according to their infectious status (e.g., susceptible, infected, recovered). In the context of sexually transmitted infections, the assumption of homogeneous sexual behaviors underlying compartmental models is unrealistic, as informed by surveys showing high heterogeneity across individuals [1]. Agent-based models (ABMs) appear to be the most suitable approach to reproduce individual behaviors and evaluate their effects at the population scale [2]. Modelling co-circulation of several pathogens is possible in compartmental models [3] but is easier in ABMs, as it prevents from making strong assumptions about parameters such as duration of infection.

A. C. M. Thiébaud and L. Opatowski—are equally contributed.

We present here a new ABM combining heterogeneous mixing through a human contact network and between-genotypes interactions. We applied it to the simulation of between-humans transmissions of genital infections with human papillomavirus (HPV) in the French context [4]. HPV infections are among the most common sexually transmitted infections, especially in younger people, and are causal agents for cervical cancer [5, 6]. Two vaccines, protecting against a fraction of HPV genotypes only, were introduced in France in 2007. As a consequence, anticipating vaccination impact on HPV infection dynamics at the population level is crucial but it requires integrating assumptions about how circulating genotypes interact for host infection. Empirical evidence suggests that within-host infection by two genotypes may affect each genotype's viral load, cell-infection ability or infection duration [7, 8]. These interactions could influence vaccination impact, as observed with pneumococcal vaccination [9].

2 Related Works

The only few models of HPV transmission which have formalized potential interactions between genotypes assumed homogeneous sexual contacts [7, 10–12]. Because sexual behaviors affect the risk of infection, it is likely that their diversity has a marked impact on co-circulation of HPV genotypes at the population level [13]. Therefore, faithful replication of sexual behaviors is essential to correctly interpret observed HPV prevalence and provide more accurate projections of the ecological consequences of vaccination. Yet, the ABMs so far developed to study HPV vaccine and/or screening effectiveness have not considered between-genotypes interactions [14–18].

3 Methods

We developed in C++ (version 4.9.0) a multi-agent system to formalize HPV genotype transmission processes in the population. The program includes two components: (1) at the agent level, to define individual characteristics and related processes; (2) at the population level, to define parameters of the agent's environment and to initiate processes for all agents at each time step.

3.1 Overview of the Model

Purpose. The ABM simulates a realistic population of individuals in heterosexual partnership and co-circulation of distinct HPV genotypes in interaction. A vaccination campaign is also simulated, assuming that vaccinated individuals are protected against a portion of genotypes (called “vaccine types”, VT) whereas their infectious risk to the rest of the circulating types (called “non-vaccine types”, NVT) is not altered. The main goal is to understand how vaccination affects global infection dynamics in the presence of between-genotypes interactions on a heterogeneous partnership network.

State Variables and Scales. The agents are human individuals characterized by a set of variables listed in Table 1 and described below. Each agent is explicitly modeled and characterized by his/her sex, age, sexual activity (number of partners he/she will have

during the year), partnership status (virgin, with a partner, or single), vaccination status, and infection status for each genotype g (susceptible to infection with g , infected with g , naturally immune to g , or vaccinated if g is a VT).

Table 1. Individual variables characterizing each agent

Variable	Type	Update time
ID	Code number	At entrance
Sex	Boolean	At entrance
Age	Number weeks	At each time step
Vaccination status	Boolean	At entrance
Variables related to partnership process		
Sexual activity class	Integer	At 15, 17, 19 and 24 years
No partner during the current year	Boolean	On the first day of each year
Partnership status	String	At the beginning/end of partnership
Partner ID	Pointer	At the beginning/end of partnership
End date of partnership or inactivity	Simulation week number	At the beginning/end of partnership
Counter of partners during the current year	Integer	At each new partnership; reset of each year
Counter of partners over the simulation	Integer	At the onset of new partnership
Variables related to the infection process – for each genotype g		
Infection status	String	At the beginning/end of infection/immunity
End date for infection	Simulation week number	At the onset of infection
Natural immunity status	String	At the end of infection
End date for natural immunity	Simulation week number	At the end of infection
Counter of infections over the simulation	Integer	At the onset of infection

Environmental Variables. Agents share a number of environmental variables characterizing the population as presented in Table 2 and detailed in the following sections. In most countries, HPV vaccination was first recommended for females before the age of 15 years, with no males vaccination [19]. We thus assumed here that only females were vaccinated, and that vaccination occurred before their entrance in the model. Agents are not explicitly distributed in space.

Initialization. Model initialization is achieved in successive steps. First, all environmental variables are initialized, as detailed in Table 2. Then all agents are created, with their partnership and infection statuses and counters initialized. At initialization, sex and age of each agent are randomly drawn to ensure homogeneous distribution of individuals across sex and age. Sexual activity and age at first partnership are drawn from probabilistic distributions (if the agent is older than his/her age at first partnership, he/she can directly look for a partner). Infection status with respect to VT and NVT is randomly drawn according to the probabilities of infection and co-infection defined for initialization (Table 2). Those initial probabilities are set to arbitrary non-null values without any impact on prevalence equilibrium.

Process Overview and Scheduling. The model proceeds in several processes which are sequentially updated on a weekly or yearly basis, depending on the processes, as detailed below. Figure 1 summarizes the execution of the three major weekly processes: demographic, partnership and infection processes.

Table 2. Environmental variables characterizing the population and their values

Description	Default value	Rationale
Population size (N)	800,000 individuals	For computational purposes
Population age range (in years)	From 15 to 30	Fixed
Variables related to partnership process		
Age at first partnership (in years)	Normal distribution with mean 17.5 in females ($sd^1 = 3.8$), 16.8 in male ($sd^1 = 4.0$)	Mean, calibrated on [1]
Sexual activity class	According to sex and age ²	From [1] and calibrated
Sexual inactivity (no partner) during the year	According to sex and age ²	From [1] and fixed
Duration of partnership	According to sexual activity and age ²	Mean, from [1] and calibrated
Duration without partner	According to sexual activity ²	Mean, calibrated
Frequency of sexual intercourse within a partnership	2 per week	Fixed [1]
Duration of partner search before mixing between sexual activity groups	According to sexual activity ²	Calibrated
Maximum proportion of individuals changing from one extreme class to another at ages 17, 19 and 24 years	95% for females and 45% for males	Calibrated
Variables related to infection process		
Number of genotypes	2 vaccine types (VT) 12 non-vaccine types (NVT)	According to genotypes reported in [20]
Transmission probability for VT and NVT genotypes	$\beta_V = 0.16$ per sexual act (VT) $\beta_{NV} = 0.125$ per sexual act (NVT)	Calibrated
Duration of infection (in weeks)	Exponential distribution with mean 52	From [21]
Duration of immunity (in weeks)	Exponential distribution with mean 12	Calibrated
Strength of between-genotypes interaction (γ)	1	To be varied
Initial probability of infection	0.2 for NVT and 0.08 for VT	Fixed
Initial probability of co-infection among those initially infected	0.4 for any first genotype g	Fixed
Date of vaccine introduction	After 70 years of simulation	To ensure prevalence equilibrium
Vaccine coverage	60% of females by age cohort	To be varied

¹Standard deviation; ² Detailed in appendix Table 3

Demographic Process. Agents age one week with every time step. On the day they turn 30 years, agents leave the model. When an agent exits the population, he/she is directly replaced by a new 15-year-old individual with an equal probability of being male or female, to ensure the population remains stable over time. At inclusion the agent's states and counters are initialized. Sexual activity and date of first partnership are drawn from

probabilistic distributions (see Table 2). In addition, if a vaccination campaign is in progress at a female's entry date, she will be vaccinated or not according to vaccine coverage, protecting her against VT with 100% efficacy for at least 15 years.

Partnership Process. Every week, the list of agents available for a new relationship is updated; each agent from this list then searches for a new partner according to his/her sexual activity and age. If the agent has been waiting on the list for a predefined period of time without finding a partner with similar sexual activity, he/she is allowed to search any others. If a partner is found, partnership duration is drawn according to the agent's sexual activity and age (Table 3 in appendix). When a partnership is set, the two partners involved are assumed to be in contacts at an arbitrary frequency of two sexual intercourses per week over the partnership duration. When the partnership end date is reached, the partnership is split. For each of the two agents involved, inactivity duration is drawn according to his/her sexual activity, defining the date for the agent to become available again for a new partnership. Simultaneous partnership is not allowed.

Infection Process and Disease Natural History. At each time step, the model first proceeds with infection transmission. For each genotype g , all partnerships characterized by one partner infected with g and one partner susceptible to g (*i.e.* is not already infected with g nor naturally immunized to g) are evaluated: transmission of g from the infected to the susceptible partner occurs with a transmission probability parameter β_V or β_{NV} depending on whether genotype g is VT or NVT, respectively. If an infection occurs, infection duration is drawn from an exponential distribution (Table 2). Second, the model proceeds with infection status update for all infected agents. For any genotype g , all agents infected with g who reach their recovery date change their infection status: they become immune to g and cannot transmit g anymore. Acquired natural immunity is assumed to confer total protection against g for a limited duration which is drawn from an exponential distribution (Table 2). All immunized agents reaching the end date of their immunized status become fully susceptible to g again.

Other Annually Scheduled Processes. Processes are executed the first week of each year in the following order. First, counters of cumulative numbers of partners over the current year are set to 0 for all agents. Second, a portion of agents aged 17, 19 and 24 years and not engaged in partnerships > 1 year are randomly selected to change sexual activity to conform to changing sexual behavior with age. Changing from one extreme behavior to the other is limited to a proportion of individuals per sex (Table 2) to favor milder transitions. Numbers are drawn to match the targeted distribution of sexual activity classes by age category. Moreover, in all sexual activity classes, a fraction of agents who will have no partner over the coming year is randomly selected (Table 2).

3.2 Design Concept

Within-Agent Ecological Genotypic Interaction. When two genotypes simultaneously infect an agent, interaction between genotypes may occur. Interactions are defined unidirectionally as the VT presence affect the duration of NVT infection. Thus, we assume that, if an agent already infected with a vaccine genotype g_V is subsequently infected with a non-vaccine genotype g_{NV} , the average duration of infection with g_{NV}

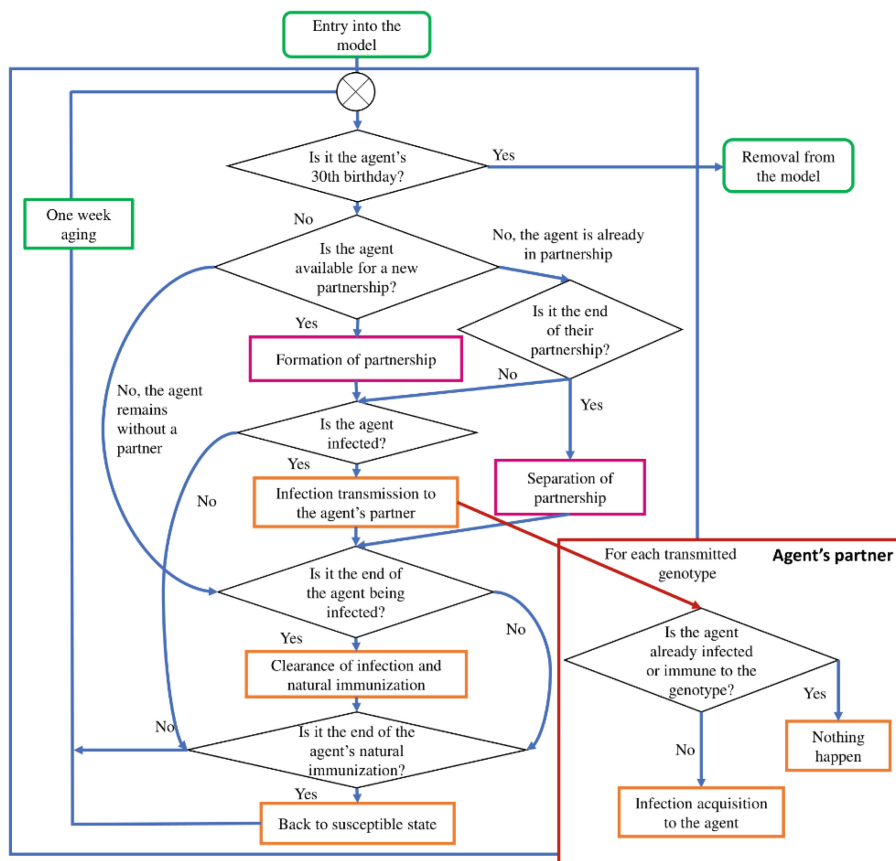


Fig. 1. Schematic representation of the ordered weekly processes that could occur for each agent under conditions. In green the demographic process, in pink the partnership process and in orange the infection process.

is modified by a multiplicative factor γ , called the strength of interaction. Both competitive, whereby $\gamma < 1$, *i.e.*, the presence of VT reduces duration of infection with NVT, and synergistic interaction, whereby $\gamma > 1$, *i.e.*, the presence of VT enhances the duration of infection with NVT, are allowed. Since the infection period coincides with the possible transmission period, this also has impact on the length of time the infected agent can transmit g_{NV} .

Stochasticity. The model is stochastic both through the definition of random parameters and through random processes. Sexual activity, age at first partnership, age difference between partners, and all durations (of partnership, of inactivity between two partnerships, of infection, of immunization after vaccine introduction) are randomly drawn from probability distributions (Table 2). Finding a partner, changing sexual activity and getting infected are defined as random processes.

Observations. A large variety of data can be collected from model simulations. Here, we simply aggregated agent data to obtain numbers of infected female by age group.

4 Results

4.1 Data

Partnerships. We used data from a French population-based cross-sectional survey [1] to define parameters characterizing the partnership process and calibrated the model with the least-squares distance minimization method to reproduce realistic contact patterns. We notably used distributions of age at first partnership, age difference between partners, and number of partners, duration of partnership, and duration without partner by age categories (18–19, 20–24 and 25–29 years).

Infections. Because the proportions of infected individuals (prevalence) were not available by age and genotype in the pre-vaccine era for the French general population, we used distributions reported from an epidemiological study carried out in the United States [20], a country in which HPV epidemiology has been suggested to be similar to that in France [6]. From [20] we extracted VT prevalence (gathering 2 genotypes: HPV 16 and 18) and NVT (gathering 12 genotypes: HPV 31, 33, 35, 39, 45, 51, 52, 56, 58, 59, 66 and 68) in the pre-vaccine era for females by age group (14–19, 20–24 and 25–29 years). We calibrated transmission probability parameters β_V and β_{NV} and the mean duration of immunity for all genotypes assuming no genotypic interaction (strength $\gamma = 1$). For each other interaction scenario, β_{NV} was calibrated again so that all scenarios fit the same observed pre-vaccine NVT prevalence.

4.2 Input and Simulations

The model was initialized with 800,000 interconnected agents. We experimented 7 interaction scenarios ($\gamma = 0.5, 0.7, 0.9, 1, 1.1, 1.3, 1.5$) with 10 simulations each. The explored genotypic interaction values and the corresponding transmission probability parameters were given as input to the model. First simulations were run for 70 years to reach prevalence equilibrium in the pre-vaccination period. Then vaccination was introduced at time $t = 0$, assuming 60% vaccine coverage. The model was simulated for 40 years following vaccine introduction. Simulations of independent iterations of each scenario over 110 years were run in parallel and took about 2 h on the computational and storage services (TARS cluster) provided by the IT Department at Institut Pasteur, Paris. Statistical analyses and graphics were computed using R (version 3.5.2). Sensitivity analyses were also carried out to evaluate the impact of our model assumptions on observed results.

4.3 Output

Figure 2 displays the VT and NVT prevalence dynamics before and after vaccine introduction for 10 simulations in seven distinct scenarios of genotypic interaction strength.

Vaccine Impact on VT Prevalence. Before vaccine introduction, the model was able to reproduce realistic VT prevalence patterns, with 10.7% infected females on average. Prevalence was slightly higher in 20–24-year-olds than in other age groups and lowest in 15–19-year-olds. After vaccine introduction, an expected decrease of VT prevalence was observed, which stabilized to a new equilibrium (3.2%) once the age cohort had had a chance to be immunized (after 5 years among 15–19-year-olds up to 15 years among 25–29-year-olds). All calibrated interaction scenarios yielded undistinguishable patterns of VT prevalence and stochasticity was weak, as demonstrated by the limited variability across the 10 simulations displayed for VT prevalence in Fig. 2.

Vaccine Impact on NVT Prevalence. The model also satisfactorily reproduced NVT prevalence before vaccine introduction, reaching 26.5% infected females on average. The same pattern of NVT prevalence was seen across age groups as with VT. Minor differences were visible across calibrated interaction scenarios. Stochasticity was more pronounced for NVT than for VT, particularly in the 20–24 age group, but standard deviations remained limited to <4% in all scenarios.

After vaccine introduction, NVT prevalence trends strongly depended on the interaction scenario (Fig. 2). Without of genotypic interaction ($\gamma = 1$), NVT prevalence remained constant over time. In contrast, under competitive interaction ($\gamma < 1$), NVT prevalence increased while, conversely, under synergistic interaction ($\gamma > 1$), NVT prevalence decreased over time until reaching a new equilibrium. The magnitude of prevalence variations depended on the strength of interaction: the more it deviated from 1, the more NVT prevalence deviated from pre-vaccine equilibrium. The increase or decrease magnitude also appeared to be proportional to NVT prevalence before vaccine introduction. Moreover, variations in NVT prevalence appeared to be most pronounced in 20–24-year-old females who also had the highest prevalence before vaccine introduction. Times to reach the new prevalence equilibrium was the same for NVT and VT.

Sensitivity Analyses. With respect to the partnership process, the first 10 combinations of parameter values that best minimized the calibration criteria resulted in VT and NVT prevalence curves changing at the same time and with comparable magnitude [22]. For the infection process, changing one parameter value at a time, whether average duration of infection, average duration of immunity, or VT and NVT transmission probabilities, affected VT and NVT prevalence equilibrium quantitatively but not the overall qualitative dynamics following vaccine introduction [22].

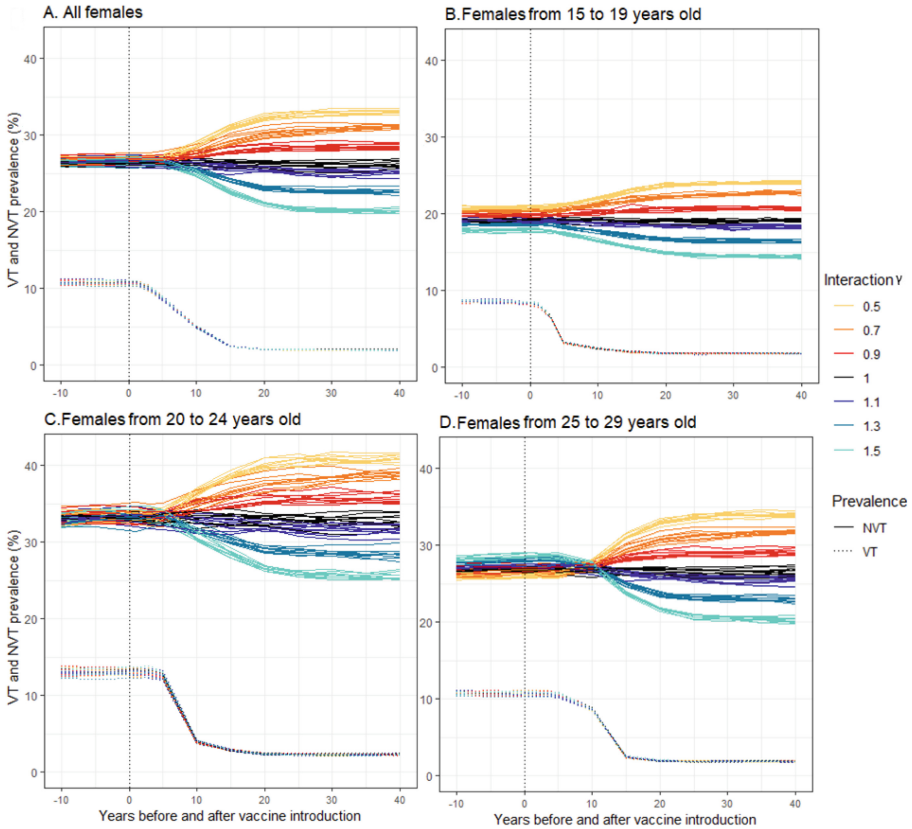


Fig. 2. Proportions of infected with NVT (upper plain line curves) or VT (lower dashed curves) over time according to γ interaction scenarios (colors, 10 simulations for each) in (A) All females; (B) Females aged 15 to 19 years; (C) Females aged 20 to 24 years; (D) Females aged 25 to 29 years. The vaccine is offered from $t = 0$ to all females <15 years of age with 60% of coverage.

5 Discussion

In this paper, we introduce a new ABM simulating the co-circulation of VT and NVT in a heterogeneous human network. The main originality of this ABM is to consider genotype interactions. The agent-based approach makes it possible to define precisely natural history of HPV infection as well as co-infections in individuals and to assess vaccine impact on the overall dynamics of HPV infection at the population level. The presented illustration is an example of how the model can be used to simulate VT and NVT prevalence according to genotypic interaction scenarios and prevention measures such as vaccination. Successively calibrated interaction scenarios help investigate how synergistic interactions could contribute to the decrease in overall HPV prevalence aimed by vaccine introduction. Conversely, we show that competitive interactions may lead to substantial increase in NVT prevalence: because VT prevalence decreases following vaccine introduction, so does the prevalence of VT and NVT coinfections, resulting in

longer duration of infection with NVT and more opportunities for NVT transmission. This ecological replacement can potentially minimize the decrease in or even increase overall HPV prevalence, hence limiting vaccine benefit at the population level. The ABM also enabled us to assess the length of follow-up and statistical power needed to detect prevalence variations in epidemiological studies [23].

Although vaccine coverage is much lower in France (around 30%), we presented here a scenario with 60% vaccine coverage to highlight the potential for genotype replacement. With lower coverage, variations in NVT prevalence were still visible under non-neutral interactions, but with reduced magnitude [4, 23]. We also chose a fixed population of 800,000 individuals, a size typical of a big metropolitan area, to obtain stable results and avoid additional complexity of modelling spatial constraints on contact mixing. Moreover, simulating the real French population size (~9,200,000 individuals 18–30 years of age) would require extremely long calculation times and would only slightly reduce the already limited variability of our results.

Despite a detailed agent-based approach, some simplifications and assumptions were made to compensate for the lack of data and gaps in knowledge. First, we could not calibrate the model to sexual behaviors reported by males and females simultaneously because distributions of numbers of partners did not match. In the CSF survey, males reported larger numbers of partners than females [1]. Reporting bias differential by sex [24] may explain part of our calibration difficulties. To overcome those difficulties, we calibrated the partnership parameters on females' data, while trying to make them consistent with males' data in terms of cumulative number of partners. Reassuringly, the cumulative distribution of the number of sexual partners obtained was close to a power law, which is characteristic of a network of sexual contacts [4]. Moreover, because our interest was to simulate realistic heterogeneous contact patterns that can reproduce the typical bell-shaped curve of HPV prevalence according to age, the model was run for decades in a preliminary step assuming unrealistically stable behaviors only to reach prevalence equilibrium before vaccine introduction. Second, to keep our model relatively simple, only two transmission probability parameters were considered for VT and NVT, thus mimicking average dynamics among those two groups. In reality, large differences in prevalence are reported not only between VT and NVT groups, but also within each group [20]. It could therefore be relevant in future work to simulate VT and NVT more finely by calibrating parameters by genotype. This would allow to study each genotype's circulation and interactions. Finally, simulation results presented here were restricted to the assumption that genotypic interaction affected the infection duration of a second acquired virus. Alternative assumptions including symmetrical VT-NVT interactions have been explored using our ABM to measure their population-level impact on prevalence and compare their credibility [4].

In conclusion, our model formalizing both sexual behavior heterogeneity and co-circulation of distinct genotypes allows us to evaluate pathogen transmission dynamics in realistic sexual networks. In the context of HPV, our simulation results show that between-genotypes interactions can significantly impact vaccination effectiveness at reducing HPV prevalence. This model can be used to further investigate the impact of most recent HPV vaccines targeting a larger number of genotypes, as well as the

extension of vaccination to males. It could easily be adapted to study other research questions related to sexually transmitted pathogens and their interactions.

Appendix

Table 3. Complementary values of variables related to the partnership process according to sexual activity class and possibly age and sex

Sex	Age (years)	Number of partners per year			
		1	2–3	≥4	0 ^a
Sexual activity class (% by sex and age category)					
Female	15–17	85.00 (calibrated) ^b	12.00 (calibrated)	3.00 (calibrated)	3.63 (fixed)
	18–19	68.99	24.72	6.29	3.63
	20–24	75.84	21.56	2.60	7.24
	25–29	85.75	11.07	3.18	5.59
Male	15–17	75.00 (calibrated)	19.00 (calibrated)	6.00 (calibrated)	14.38 (fixed)
	18–19	60.42	27.59	11.99	14.38
	20–24	61.85	28.15	10.00	10.89
	25–29	76.30	17.23	6.47	6.69
Duration of partnership					
Any	15–19	4 categories (<1, 1–4, 5–9 and > 10 years) distributed as in [1] ^c	Gamma distribution with calibrated mean 14 weeks (fixed variance 8)	Gamma distribution with calibrated mean 4 weeks (fixed variance 5) ^d	Not relevant
	20–24	4 categories distributed as in [1] ^c			
	25–29	4 categories distributed as in [1] ^c			
Duration of inactivity between two partnerships					
Any	All	Uniform distribution [1:104] weeks	Gamma distribution with calibrated mean 14 weeks (fixed variance 8)	Gamma distribution with calibrated mean 4 weeks (fixed variance 5) ^d	At least 52 weeks
Duration of partner search before mixing between sexual activity classes					
Any	All	10 weeks (calibrated)	5 weeks (calibrated)	30 weeks (calibrated)	Not relevant

^a Individuals without a partner are randomly selected each year from the population; ^b When not specified fixed or calibrated, values are extracted from [1]; ^c One category of partnership duration is drawn from a multinomial distribution whose parameters are the proportions reported in [1], then a precise value is drawn from a uniform distribution within the duration category; ^d The duration of partnership and between two partnerships are identical in most sexually active individuals and drawn only once a year

References

1. Bajos, N., Bozon, M.: Sexualité, genre et santé : les apports de l'enquête Contexte de la sexualité en France. La Découverte (2008)
2. Auchincloss, A.H., Diez Roux, A.V.: A new tool for epidemiology: the usefulness of dynamic-agent models in understanding place effects on health. *Am. J. Epidemiol.* **168**, 1–8 (2008)
3. Sanz, J., Xia, C.-Y., Meloni, S., Moreno, Y.: Dynamics of interacting diseases. *Phys. Rev. X* **4**, 041005 (2014)
4. Bonneault, M., et al.: Contact patterns and HPV-genotype interactions yield heterogeneous HPV-vaccine impacts depending on sexual behaviors: an individual-based model. *Epidemics* **39**, 100584 (2022)
5. Walboomers, J.M., et al.: Human papillomavirus is a necessary cause of invasive cervical cancer worldwide. *J. Pathol.* **189**, 12–19 (1999)
6. de Sanjosé, S., et al.: Worldwide prevalence and genotype distribution of cervical human papillomavirus DNA in women with normal cytology: a meta-analysis. *Lancet Infect. Dis.* **7**, 453–459 (2007)
7. Murall, C.L., McCann, K.S., Bauch, C.T.: Revising ecological assumptions about human papillomavirus interactions and type replacement. *J. Theor. Biol.* **350**, 98–109 (2014)
8. McLaughlin-Drubin, M.E., Meyers, C.: Evidence for the coexistence of two genital HPV types within the same host cell in vitro. *Virology* **321**, 173–180 (2004)
9. Choi, Y.H., et al.: 7-valent pneumococcal conjugate vaccination in England and Wales: is it still beneficial despite high levels of serotype replacement? *PLoS ONE* **6**, e26190 (2011)
10. Elbasha, E.H., Dasbach, E.J., Insinga, R.P.: A multi-type HPV transmission model. *Bull. Math. Biol.* **70**, 2126–2176 (2008)
11. Pons-Salort, M., et al.: Exploring individual HPV coinfections is essential to predict HPV-vaccination impact on genotype distribution: a model-based approach. *Vaccine* **31**, 1238–1245 (2013)
12. Man, I., Vänskä, S., Lehtinen, M., Bogaards, J.A.: Human papillomavirus genotype replacement: still too early to tell? *J. Infect. Dis.* **224**, 481–491 (2021)
13. Shiboski, S., Padian, N.S.: Population- and individual-based approaches to the design and analysis of epidemiologic studies of sexually transmitted disease transmission. *J. Infect. Dis.* **174**, S188–S200 (1996)
14. Olsen, J., Jepsen, M.R.: Human papillomavirus transmission and cost-effectiveness of introducing quadrivalent HPV vaccination in Denmark. *Int. J. Technol. Assess. Health Care* **26**, 183–191 (2010)
15. Van de Velde, N., et al.: Population-level impact of the bivalent, quadrivalent, and nonavalent human papillomavirus vaccines: a model-based analysis. *J. Natl. Cancer Inst.* **104**, 1712–1723 (2012)
16. Matthijsse, S.M., et al.: The role of acquired immunity in the spread of human papillomavirus (HPV): explorations with a microsimulation model. *PLoS ONE* **10**, e0116618 (2015)
17. Burger, E.A., Campos, N.G., Sy, S., Regan, C., Kim, J.J.: Health and economic benefits of single-dose HPV vaccination in a Gavi-eligible country. *Vaccine* **36**, 4823–4829 (2018)
18. Johnson, H.C., et al.: Effect of HPV vaccination and cervical cancer screening in England by ethnicity: a model-ling study. *Lancet Public Health.* **3**, e44–e51 (2018)
19. ECDC: Guidance on HPV vaccination in EU countries: focus on boys, people living with HIV and 9-valent HPV vaccine introduction. <https://www.ecdc.europa.eu/en/publications-data/guidance-hpv-vaccination-eu-focus-boys-people-living-hiv-9vHPV-vaccine>. Accessed 02 July 2020
20. Markowitz, L.E., et al.: Reduction in human papillomavirus (HPV) prevalence among young women following HPV vaccine introduction in the United States, national health and nutrition examination surveys, 2003–2010. *J. Infect. Dis.* **208**, 385–393 (2013)

21. Trottier, H., et al.: Human papillomavirus infections with multiple types and risk of cervical neoplasia. *Cancer Epidemiol. Biomark. Prev. Publ. Am. Assoc. Cancer Res. Cosponsored Am. Soc. Prev. Oncol.* **15**, 1274–1280 (2006)
22. Bonneault, M.: Modélisation dynamique des infections et co-infections génitales à papillomavirus humain (HPV) et de l'impact à long terme de la vaccination anti-HPV, Université Paris-Saclay, <https://theses.hal.science/tel-03231970> (2021)
23. Bonneault, M., et al.: Ability of epidemiological studies to monitor HPV post-vaccination dynamics: a simulation study. *Epidemiol. Infect.* **151**, e31 (2023)
24. Mitchell, K.R., et al.: Why do men report more opposite-sex sexual partners than women? analysis of the gender discrepancy in a British national probability survey. *J. Sex Res.* **56**, 1–8 (2019)



Towards the Building of a Surveillance Network for PPR-Like Diseases in Nigeria: Identifying Potential Sentinel Node in a Partially-Known Network

Asma Mesdour^{1,2(✉)}, Sandra Ijioma⁵, Muhammad-Bashir Bolajoko⁵, Elena Arsevska^{1,2}, Mamadou Ciss³, Mathieu Andraud⁴, Andrea Apolloni^{1,2}, and Eric Cardinale¹

¹ CIRAD, UMR ASTRE, INRAE, 34398 Montpellier, France

asma.mesdour@cirad.fr

² UMR ASTRE, CIRAD, Université de Montpellier, Montpellier, France

³ ISRA, LNERV BP 2057 Dakar-Hann, Senegal

⁴ ANSES, Ploufragan-Plouzané-Niort Laboratory, EPISABE Unit, Ploufragan, France

⁵ National Veterinary Research Institute, Vom, Nigeria

Abstract. Peste des Petits Ruminants (PPR) is a highly contagious disease affecting sheep and goats and spreading through livestock movements. The FAO and WOAHA have set the goal to eradicate it by 2030, with one of the key steps being the improvement of surveillance networks. In our study, we utilize an SI model to simulate the spread of PPR across three Nigerian States due to animal movements. Our objective is to identify areas that could serve as potential sentinel nodes, which are nodes that may become infected promptly at the onset of epidemics. We reconstructed the mobility network using (the partial) movement data collected through market surveys. Due to the limitations of the study, we implement plausible network modifications and generated eight distinct network configurations. For each of these configurations, we assessed the extent of the epidemics, the presence of recurrent patterns and characterize sentinel nodes. Our findings indicate that minor network modifications do not impact the final sizes of epidemics. However, removing a central market or introducing a significant number of new links can lead to noticeable shifts in epidemic propagation trajectories. Regardless of the specific configurations considered, we identified three clusters of epidemic origins, each exhibiting distinct behaviors. Nevertheless, geographical proximity among these origins does not necessarily imply similar transmission patterns. Furthermore, our study highlights that, in most configurations, attributes such as the in-H index, eigenvector and in-degree centralities emerge as the most influential factors in defining vulnerable nodes. Therefore, we recommend prioritizing data collection efforts in central markets due to their wealth of information resources

Keywords: Mobility network vulnerability · animal movement · livestock diseases

M. Ciss, M. Andraud, A. Apolloni and E. Cardinale—contributed equally.

1 Introduction

The emergence of pathogens in livestock poses one of the gravest threats to global animal health, welfare, food security, and biodiversity. Some of these destructive diseases impact national and international trade and persist as endemic in various regions [1]. Among these threats, Peste des Petits Ruminants (PPR), an airborne disease affecting mostly domestic small ruminants (goats and sheep), stands out as one of the most devastating livestock diseases. Originally documented in Cote d'Ivoire in 1942, PPR is now present in most of the African and Asian countries and recently has appeared at the gates of Europe [2]. The disease's geographic extension can be attributed to the rapid animal movements [3]. The high case-fatality rate (around 30%) induces annual economic losses estimated at USD 2.1 billion, threatening the livelihoods, food security, and employment opportunities of approximately 300 million families (FAO, 2023). These facts, along with the existence of an effective vaccine, have driven the FAO and WOA to target PPR for eradication plan by 2030. One of the steps of the eradication program aims at improving surveillance systems on the circulation of the virus [4].

PPR diffusion can be described as a propagation over a complex network where nodes represent locations and links represent animal exchanges between two of them. From a complex network point of view, defining an effective surveillance system can be translated to finding a set of nodes, called *sentinel nodes*, able to provide accurate and timely insights into epidemic dynamics [4], *i.e.*, nodes that will be affected often and before the peak of the epidemic. For a node, the probability of getting infected depends on several factors, among them the epidemiological characteristics of the pathogen, the structure of the network (*i.e.*, the position of the node in it), and where the epidemic originates (*seed*). However, identifying *sentinel nodes* can be proven difficult in practice: centrality measures can greatly fluctuate over time and depend on details of the system; they can also be difficult to link to interpretable characteristics [5]. In practice, the impact of surveillance and control strategies depends strongly on the quality of the data used [5]. Detailed animal movement could be used to inform highly detailed models and identify recurrent patterns more easily [6]. Limited resolution data can make it difficult to identify *sentinel nodes* correctly [7]. Regrettably, in countries particularly susceptible to PPR, data collection remains infrequent and limited to specific regions over limited time. In Nigeria, livestock identification system, is not implemented. Therefore, information on animal mobility can only be collected through ad-hoc activities, like market surveys. However, these activities are time and resource consuming, restricted to specific areas and in a specific time of the year, and the number of respondents is generally relatively small. Because of these limitations, information collected is fragmented and provides a partial representation of the mobility network.

In this work we use incomplete movement data from market surveys in 3 Nigerian States to identify a set of possible *sentinel nodes*. Uncertainty analysis was therefore performed through simulations of modifications of the network topology, to analyze how the incompleteness of the data impact epidemic's extension. Moreover, we studied the role that *seed nodes* (area where first cases occurred) could have on the propagation and identified possible clusters of seed nodes that could have similar transmission patterns. Finally, we identify the structural characteristics of *sentinel nodes*.

2 Material and Methods

2.1 Data and Epidemic Simulation

Nigeria is a federal country with three administrative divisions: State, Local Government Area (LGA), District/Ward. In this study we use domestic small ruminants (goats and sheep) mobility data collected through survey conducted in 10 markets in 3 States in the central and northeastern part of Nigeria: 6 in Plateau, 2 in Bauchi State and 2 in Kano State.

In each market around 100 livestock owners/ traders were questioned about: origin/destination of the movements (State, LGA, District and village name), heads number by species, and reason for the movement. To avoid problems related to the misspelling of villages, data were aggregated at district level (nodes of the network). A link between two districts was considered if at least a single animal was moved between the two. The result consists of an oriented network with 144 nodes and 268 links forming a single weakly connected component and 2 strong ones. From now on we indicate the network reconstructed from the data as “observed network” and will be used as reference (configuration A) in the next steps. In this work we used the following centrality measures to characterize each node: In/out Degree [8] Betweenness, Closeness [8], H index [9], Neighborhood [10] and Eigenvector centrality [11].

We simulated the propagation of epidemic through livestock movements, using an SI stochastic model. At the beginning of the epidemic all the nodes are susceptible (S) except one (the *seed* from now on) in the infected state (I), chosen among all nodes with non-null outdegree. Indicating with ρ the probability that an infected district could infect a susceptible one through animal movements, and with I_i the number of the infected neighbors of node i [12], the probability of a susceptible node i to become infected (I) follows a binomial distribution, with the probability P_i defined as follows:

$$P_i = 1 - (1 - \rho)^{I_i}.$$

In our case each time step corresponds to one week, corresponding to the mode of the record frequency for all the districts in the dataset.

Following the work by [4], we define *sentinel nodes* as nodes that are *frequently* and *timely* infected during epidemics, *i.e.*, the nodes that are most often infected, and before the peak of incidence is reached. Three factors could affect the propagation of the pathogen and the characteristics of the *sentinel nodes*: the transmission probability; the structure of the network itself; and the origin of the epidemic itself (seed). To evaluate the impact of transmission probabilities on the epidemic process, we considered five different values of ρ varying from 0.01 to 0.75 (0.01, 0.1, 0.3, 0.5, 0.75). For each value of ρ , we examined various network configurations (as described in the following section) and initiated the epidemic by selecting a seed node from those with an outdegree non-null. 100 epidemics simulations were performed on the observed graph and then on each modified network, simulations were stopped when no new infected node was recorded for 10 consecutive time steps.

In our analysis we inferred the role of missing links and nodes in the propagation of disease and the identification of *sentinel nodes*. Our assumption is that information is missing, and the observed network (configuration A) is just a subset of the “entire”

network. To account for the missing information of the network structures (partially observed) and fill the gap in our knowledge, we run several scenarios with modified networks. This will help to assess the impact of modifications of network structure on the epidemic spreading in comparison to results from the observed one. Moreover, this will help identifying movements having an impact on the propagation whose existence/absence should be confirmed by field activities. Nevertheless, exploring all scenarios could be a tantamount effort and, in this work, we focus on some scenarios that could be compatible to constraints in data collection and in-field situations. We considered two categories of modifications to assess kinds of error/missing information:

- i) **Modifications due to Structural Misinformation:** Respondents could provide partial information based on their experience. Furthermore, not all the traders present in the market were surveyed. We mimic the modification of trade relationships through random permutations of links or completion of observed data.
 1. **Random Partial Reorganization:** Randomly reorder 5% (10 links; configuration B1) and 40% (58 links; configuration B2) of network links while maintaining the same node degrees;
 2. **Random Link Insertion:** Randomly add 5% (configuration C1) and then 40% (configuration C2) of network links
- ii) **Modifications due to Survey Limitation:** due to resources limitedness not all markets could be sampled, thus reducing information about nodes and links present in the network.
 3. **Central Market Removal:** Remove the most connected market located in Plateau State (configuration D1), and then remove the least connected market (configuration D2);
 4. **Peripheral Market Removal:** Remove one of the surveyed markets located in Kano (configuration E1), and another surveyed market located in Bauchi State (configuration E2)

3 Seeds' Cluster Detection

Following a procedure similar to the one introduced in [6], we aimed to identify subsets of seeds (*seed clusters*), for which epidemics have similar behaviors in terms of size and identity of infected nodes. For each network configuration and each seed, we reconstructed the “probabilistic transmission pathway” through the following steps:

1. given a node j infected at time t , we identified all its potential infectors, *i.e.*, node's j neighbors which were previously infected;
2. an oriented link was drawn going from each potential infector to infected node j ;
3. Steps 1 and 2 were repeated over all simulations and using different transmission probabilities. The weight to each link in the transmission pathway is associated with the frequency a link appeared in all the simulations.

The result for each seed was an oriented weighted network called probabilistic oriented pathway. Probabilistic transmission pathways for different seeds were compared

using the weighted Jaccard index [13] to gauge the similarity between them and used as distance measure. The optimal number of clusters was first identified using the *elbow method* and then k-means classification was applied. The clusters detected in the reference configuration (configuration A) and the modified ones, were then compared using the Rand index [14] to assess to which extent modifications could alter propagation patterns. In a second step, the number of nodes that were reached by seeds in each cluster were compared using analysis of variance, followed by a post-hoc Tuckey test, to identify statistically distinct groups. This method allowed for identification of clusters of seed which, once infected, could give origin to large or small epidemics.

4 Node's Vulnerability and Definition of Sentinel Nodes

In this work we defined *sentinel nodes* based on their vulnerability and the time at which they are infected. A univocal definition of node's vulnerability doesn't exist. We defined vulnerability as the probability of a node getting infected early in the epidemics. For each configuration and transmission probability we used simulation results to identify vulnerable nodes:

1. For each node we estimated the number of times its infection occurred before the epidemic peak (*frequency*);
2. We estimated the average frequency and the standard deviation;
3. Classical threshold (average frequency +2 standard deviations) was used to classify nodes [16];
4. Nodes with frequency higher than threshold were classified as Vulnerable Otherwise were classified as Infected (*i.e.*, after the epidemic peak); Not infected (remaining susceptible throughout the simulation process).

We then used random forest algorithm [17] to classify the centrality measures characterizing *sentinel nodes*. Moreover, we investigated whether the characteristics of the vulnerable nodes would change or not depending on the origin of the epidemics by repeating this step for each configuration.

5 Results

The Impact of Structural Variations on Final Size

In the observed network there are 63 districts with a positive out-degree. The same number is found in configurations B1, and B2. However, this number rises to 66 and 76 (configurations C1 and C2, respectively) diminishes to 60, 62, 57, and 58 (configurations D1, D2, E1, and E2 respectively). A large portion of the seeds (ranging from 39% to 43%) is concentrated in Plateau State, with a further portion (between 22% and 32%) in Bauchi State, and a comparatively smaller proportion (between 5% and 11%) in Kano State (Table 1).

Table 1. Number of nodes (districts) in all configurations taken as seed per States: A: observed network, B1: modified network with random reallocation of 5% of links, B2: modified network with random reallocation of 40% of links, C1: modified network with addition of 5% of links, C2: modified network with addition of 40% of links, D1: modified network with deletion of the most connected central market, D2: modified network with the deletion of the less connected central market, E1: modified network with deletion a peripheral market of Kano, E2: modified network with random deletion of a peripheral market in Bauchi.

State	A	B1	B2	C1	C2	D1	D2	E1	E2
Plateau's seed	26 (41%)	26 (41%)	25 (40%)	26 (39%)	26 (34%)	24 (40%)	24 (39%)	24 (42%)	25 (43%)
Bauchi's seed	18 (29%)	18 (29%)	18 (29%)	18 (27%)	18 (24%)	17 (28%)	18 (29%)	18 (32%)	13 (22%)
Kano seed's	6 (10%)	6 (10%)	6 (10%)	7 (11%)	7 (9%)	6 (10%)	6 (10%)	3 (5%)	6 (10%)
Others	13 (20%)	13 (20%)	14 (21%)	15 (23%)	25 (33%)	13 (22%)	14 (22%)	9 (21%)	14 (25%)
Total	63	63	63	66	76	60	62	57	58

Figure 1 gives an overview of the simulation results obtained for all configurations, focusing on the distribution of final sizes and the duration of epidemics - measured as the time taken to reach the maximum number of new infections. Independently of the configuration chosen, as expected, the final size and the duration of the epidemic are influenced by the transmission probability, its increase causes larger epidemics in a shorter amount of time. For all configurations the same behavior is observed for the lowest value of the transmission probability. However, when the transmission probability increases structural variations affect epidemic size. The addition of a large number of links C2 leads to a considerable expansion in the final size of the epidemic.

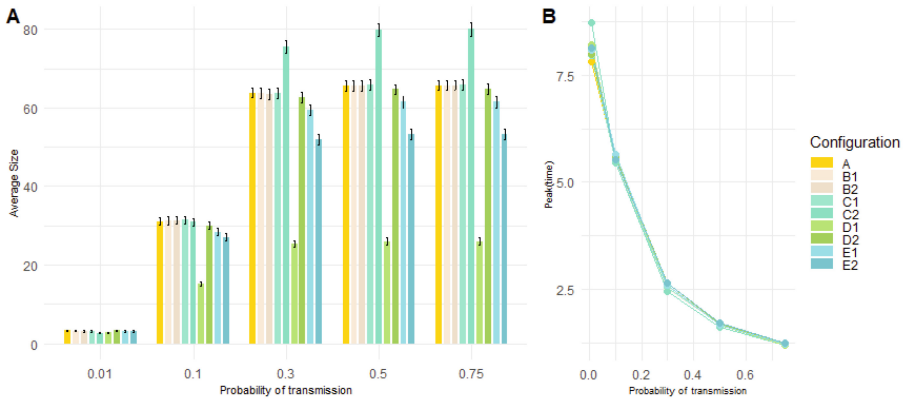


Fig. 1. Results of PPR simulations on the animal mobility network: A: Average final size in simulated epidemics under different configurations and for each transmission probability (ρ). B: Time at which the peak is reached.

Furthermore, the choice of excluding central (D1) or peripheral markets (E1, E2) provides widely different estimates of the final size of the epidemics (Fig. 1 A) already

for low values of the transmission probability ($\rho = 0.1$) but it becomes evident for larger values. Nevertheless, the time of the epidemics (Fig. 1 B) is not affected by structural changes.

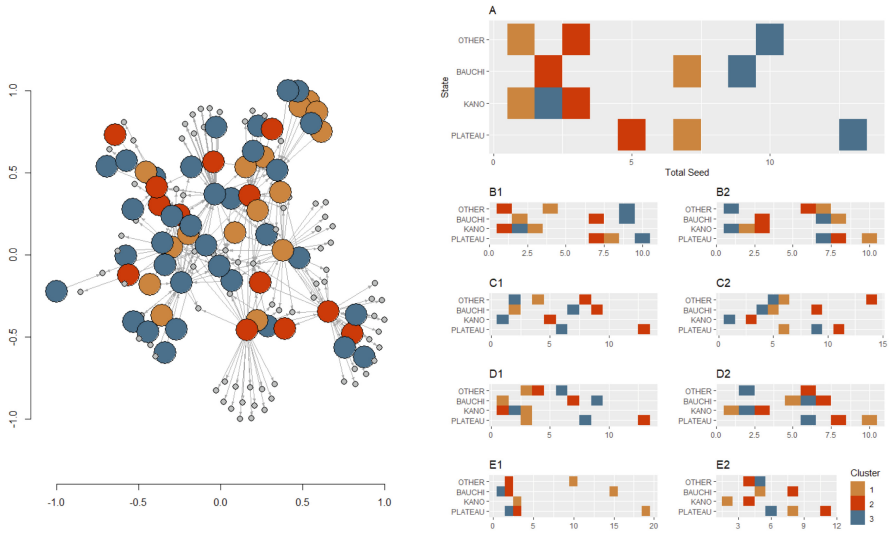


Fig. 2. Observed network where nodes are colored depending on the cluster seed they belong to corresponds to *seed cluster*; B: Number of seeds in each *seed cluster* by State. On the x-axis the number of seeds. Each line corresponds to the distribution in clusters (color) of seeds located in the States.

Seeds Clusters are Geographically Scattered:

In our analysis, each cluster consists of seeds that share a relatively similar path of invasion and whose epidemics have comparable final sizes: if epidemics were to originate from nodes belonging to the cluster, they would likely impact the same set of nodes.

For each configuration, the k-means method identifies three clusters (Fig. 2) (see more details in supplementary material). In all configurations, clusters are composed of seeds that are geographically scattered; each cluster is formed by seeds from different States (Fig. 2).

A comparison between cluster distributions in the observed graph and modified graphs could highlight the degree of sensibility of these clusters against network modifications. For small variations of link distribution, B1 configuration, the similarity is high Rand index = 0.81) and decreases when a large fraction of links is rewired configuration B2, (Rand index = 0.69). The similarity is lower when contrasted with graph configurations C1, D2, and E2 (0.56). Adding 40% of links (configuration C2) and the removal of a peripheral market (configuration E1) result in notable changes in identified clusters (0.30, 0.37 respectively). In cases where the Zawan market, the most connected, is eliminated (configuration D1), the resulting Rand index is merely 0.11.

Analysis of the epidemics sizes reveals a significant influence of the “cluster” factor in all configurations, (p-values < 0.0001) indicating that there are significant statistical differences among clusters with regards to their epidemic potential: depending on the cluster the seed belong too epidemics could have different extents. To assess the significant differences between cluster’s epidemic potentials, we used the Tukey test, comparing clusters distributions in each configuration. Results of the test were statistically significant for all configurations, indicating that some clusters tend to infect more nodes than others (see more details in supplementary material). In the following we indicate with “*propagator cluster*”, the cluster whose epidemics originating in one of it seeds reaches the largest number of nodes.

In the observed network, there are only 13 seeds forming the “*propagator cluster*”. Most seeds in the propagator cluster belong to Plateau State, followed by Bauchi, while only a small number are in Kano or other States. In configurations C1 and C2, the number of seeds in the propagator cluster increases to 25 and 22, respectively. Conversely, in configurations D1 and E1, the size of the propagator clusters significantly decreases, reaching just 3 seeds in both cases. Some configurations share common seeds within their propagator clusters. Figure 3 illustrates sets of shared seeds among different configuration sets. For example, configuration A tends to share more common seeds

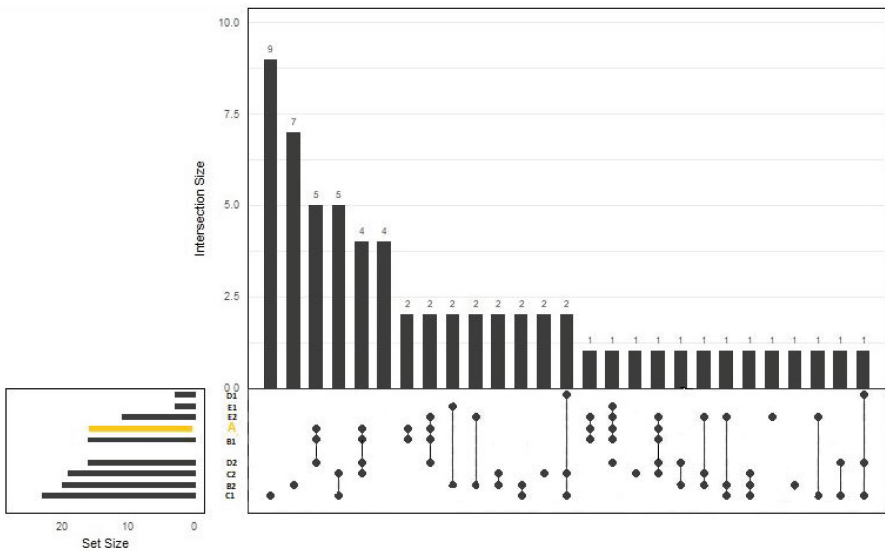


Fig. 3. Visualization of Common Seed Sets of the *propagator cluster* among Different Configuration Combinations using UpSet Diagram. The length of each horizontal bar (set) indicates the number of seeds present in a specific configuration. Vertical Bars (Intersections): represent shared seed sets that occur across different configurations. The height of each vertical bar indicates the number of seeds shared among the configurations. The intersections of the horizontal and vertical bars highlight the seeds that are common between the configurations involved.

with configurations B1, B2, C1, and E2, while having fewer in common with other configurations.

Sentinel Nodes are Those with Highest in-H index and/or In-degree and/or Eigenvector:

We used the random forest technique to classify the centrality measures characterizing *sentinel nodes*. Table 2 presents the GINI index, highlighting the most important centrality measures across all configurations within each *seed cluster* identified before. Notably, the most pivotal characteristics distinguishing vulnerable nodes within the observed network were in-H index (100%), eigenvector (80%), and in-degree (60%) for Cluster 1. While these attributes remained consistent across other clusters within the same graph, as well as across various configurations, the differentiating factor laid in the prioritization of these attributes. A unique instance was observed in Cluster 3 of Configuration E2, where betweenness is the most prominent characteristic.

Table 2. Characteristics of vulnerable nodes by cluster seed and configuration. Each column represents a configuration, and each row centrality measures per cluster. The most important GINI measures are shown below

		Gini Index (%) per Cluster and per Configuration									
Seed's Cluster	Centrality measure	A	B1	B2	C1	C2	D1	D2	E1	E2	
1	In-H index	100	100	70	60	100	>10	75	>100	22	
	Eigenvector	80	90	68	50	60	>10	58	20	40	
	In-degree	60	99	100	100	98	100	100	100	100	
	Out-closness	>10	>10	>10	>10	>10	78	>10	45	>10	
2	In-H index	100	38	35	60	60	40	38	50	40	
	Eigenvector	100	40	40	40	40	80	40	80	60	
	In-degree	90	100	100	100	100	100	100	100	100	
3	In-H index	50	85	15	>10	20	>10	>10	35	>10	
	Eigenvector	35	100	40	50	15	25	55	50	20	
	In-degree	100	80	100	100	100	100	100	100	60	
	In-Neighborhood	>10	15	40	55	>10	60	40	20	>10	
	Out-closness	>10	>10	>10	50	30	30	20	>10	>10	
	Betweenness	>10	>10	>10	>10	>10	>10	>10	>10	100	

6 Discussion

In Nigeria, the absence of a livestock identification system and automatic data centralization system hinders our possibility of depicting a reliable network of commercial livestock mobility [18]. This data would be essential to pinpoint *sentinel nodes*, which proves valuable in designing surveillance and control programs. This study was primarily focused on discerning nodes prone to be used as sentinel across three Nigerian States,

based on their intrinsic characteristics under a variety of scenarios. Despite the limitations of the observed network, which represents only a small part of all movements, this study has enabled us to investigate the effect of missing information over the propagation of diseases.

Employing elementary link operations such as link rewiring and addition, and targeted node deletion, we generated eight distinct configurations (B1 to E2). The impact of structural modifications on the dissemination of epidemics clearly becomes apparent when 40% of links are added. However, the elimination of peripheral market in Kano State, Wudil market, influence the diffusion dynamics. This finding resonates with the insights offered by Wright [19], whose research highlighted the pivotal role of peripheral nodes in the intricate dynamics of directed networks. The impact becomes even more significant when eliminating the most highly connected central markets (Zawan market). This results in a considerable reduction in the final size of the epidemic, causing a disproportionate disruption of epidemic diffusion patterns. Moreover, Zawan was one of the surveyed markets and data collected provided comprehensive insights into the network's structure and dynamics and aid in more accurate epidemic assessments. Because of this the choice of markets to be sampled should be made correctly to avoid losing vital information for surveillance and control.

Seeds Clustering:

Epidemic trajectories have been clustered based on their origins. Three clusters have been identified for each configuration whose members are consistent (high Rand Index). This indicates that these seed groups share similar transmission pathways despite network modification. Like the study of Bajardi [6], our seed clustering results reveal that geographical proximity between two nodes does not necessarily imply that they will lead to similar invasion paths.

However, removing the most central market significantly alters seed clustering, leading to behavioral changes of certain nodes. Notably, the suppression of peripheral nodes in Bauchi State triggers a substantial shift in the transmission paths of specific seeds. Some nodes alter their behavior and emerge as more influential *propagators*. This phenomenon highlights the intricate relationship between network topology and disease spread: the effect of localized changes could ripple through the system, resulting in pronounced shifts in transmission patterns. Additionally, independently of the configuration considered, these *propagator seeds* are mostly located in the Plateau State. This result can be explained by the position and the economic activity of Plateau State, the gateway for animal movements towards the densely populated area on the coast [20]. A more detailed analysis, considering several socio-economic factors, should be done to better describe the role of Plateau in comparison with other Nigerian States. Few *propagator seeds* were detected in Kano. However, this result should be interpreted cautiously, as it might be attributed to non-homogeneous sampling, (only 2 markets were sampled in both Bauchi and Kano, while 6 in Plateau). The presence of few *propagator seeds* in Kano may suggest the tendency of movements to be more important with North Africa. It is worth noticing that is the entry point in Nigeria of the “via trans-africaine 2” an international road connecting Algeria to Nigeria. The lack of this data may have underestimated Kano's role in the propagation of disease.

Identifying Sentinel Nodes:

The study successfully pinpointed around ten vulnerable nodes, taking into consideration transmission probabilities and different *seed clusters*, dispersed in 6 different States: Alaba (Lagos State), Dengi (Plateau state), Josjarawa (Plateau State), Lafia (Nasarawa State), Obolloafor (Enugu State), Okoamako (Delta State), Pali (Bauchi State), Shendam (Plateau State), Wase (Plateau State), Zawan (Plateau State). Moreover, 3 out of 6 of these States are in the south of the country (Lagos: southwestern, Delta: south, Enugu: southeast). In Nigeria, livestock production and trade have a seasonal character, with animals moving towards greener areas where resources are available, sold when resources are exploited to cover for household needs, or during the seeding and harvesting period. In general, this calendar depends on the rainy season, whose length decreases from south to north. This may suggest that there are more movements of small ruminants from north to south, looking for water and feed for small ruminants, particularly during the dry season [21]. The remaining three States are situated in the central and northern region. The central region is the least populated and least developed area of Nigeria [20]. Due to the unfavorable soil and climate conditions, it is believed that these regions serve as transit areas for small ruminants that move from the north to the more economically prosperous and vegetation-rich southern regions. Despite being one of the most densely populated State in the country, thus with a high demand of red meat, Kano does not seem to exhibit vulnerability to the disease, as none of its districts have been identified as susceptible. To better highlight the role played by Kano, we must add the international movements that tend to enter or leave through the north of the country.

Vulnerable nodes primarily stand out due to their high in-degree, in-H index, and/or eigenvector values. Notably, in-degree played a pivotal role in constructing our SI model, thus justifying its relevance as a significant feature. Analyses conducted by Herrera [4] and Colman [22] have already affirmed the ability of eigenvector centrality to identify *sentinel nodes*. By pinpointing nodes with high eigenvector centrality, we are precisely targeting these influential hubs within the network. The in-H index [9] plays a pivotal role in identifying *sentinel nodes*. Lü [9] suggested that the H-index can be a good compromise to quantify, in many cases, the influence of the nodes better than the degree. A node with a high in-H index could indicate that it is surrounded by neighbors representing diverse potential sources of infection, thereby increasing the likelihood of rapid infection. The identified characteristics remain generally consistent across all configurations, except in scenarios involving the removal of the peripheral market of Bauchi State, where betweenness emerges as the predominant feature. This suggests that when this peripheral node is present, the number of introduction sources is high; however, upon its elimination, the number of introduction pathways significantly decreases, making the most central nodes more likely to be affected. Furthermore, since this result is only observed within cluster 3, it also implies that the origin of the epidemic can play a crucial role in node vulnerability.

While the implementation of prevention and control strategies targeting nodes with a high indegree and/or in-H index and/or eigenvector may prove effective, the complexity of calculating these measures - particularly due to the scarcity of data and temporal considerations - means that we should not rely solely on these identified criteria. Optimal choices for *sentinel nodes* depend on several factors, including the network's structural

layout (e.g., nodal positioning) [22], the flow dynamics within the network, disease transmissibility (with less transmissible diseases posing tracking challenges), and temporal variations within the network (nodes maintaining stable contact sets are better suited for epidemic detection) [23].

In this work we considered only 2 extreme cases of network modifications: when a low (5%) and a high (40%) proportion of links were added/rewired/eliminated. It would be interesting to conduct a more systematical analysis, considering more levels of modification, to determine the critical threshold at which structural changes become critical and thereby impact the progression of disease. Furthermore, a more sophisticated model, capable of better capturing animal movements dynamics, volume and temporal aspects, supported by a larger volume of data, could be pursued in the future to ascertain whether these same characteristics will continue to justify node vulnerability. Furthermore, additional analysis is required to validate the model, comparing disease simulation outcomes on the data-based network and the reconstructed network, and cross-referencing these with biological information such as PCR data.

References

1. Tomley, F.M., Shirley, M.W.: Livestock infectious diseases and zoonoses. *Philos. Trans. R. Soc. B Biol. Sci.* **364**(1530), 2637–2642 (2009). <https://doi.org/10.1098/rstb.2009.0133>
2. Kumar, N., et al.: Peste des petits ruminants virus infection of small ruminants: a comprehensive review. *Viruses* **6**(6), 6 (2014). <https://doi.org/10.3390/v6062287>
3. Diallo, A., Taylor, W.P., Lefèvre, P.C., Provost, A.: Atténuation d’une souche de virus de la peste des petits ruminants : candidat pour un vaccin homologue vivant. *Rev. D’élevage Médecine Vét. Pays Trop.* **42**(3), 3 (1989). <https://doi.org/10.19182/remvt.8771>
4. Herrera, J.L., Srinivasan, R., Brownstein, J.S., Galvani, A.P., Meyers, L.A.: Disease surveillance on complex social networks. *Plos Comput. Biol.* **12**(7), e1004928 (2016). <https://doi.org/10.1371/journal.pcbi.1004928>
5. Ciaperoni, M., Galimberti, E., Bronchi, F., Cattuto, C., Gullo, F., Barrat, A.: Relevance of temporal cores for epidemic spread in temporal networks. *Sci. Rep.* **10**(1), 12529 (2020). <https://doi.org/10.1038/s41598-020-69464-3>
6. Bajardi, P., Barrat, A., Savini, L., Colizza, V.: Optimizing surveillance for livestock disease spreading through animal movements. *J. R. Soc. Interface* **9**(76), 2814–2825, (2012). <https://doi.org/10.1098/rsif.2012.0289>
7. Génois, M., Barrat, A.: Can co-location be used as a proxy for face-to-face contacts?. *EPJ Data Sci.* **7**(1), Art. No. 1 (2018). <https://doi.org/10.1140/epjds/s13688-018-0140-1>
8. Bell, D., Atkinson, J., Carlson, J.: Centrality measures for disease transmission networks. *Soc. Netw.* **21**, 1–21 (1999). [https://doi.org/10.1016/S03788733\(98\)00010-0](https://doi.org/10.1016/S03788733(98)00010-0)
9. Lü, L., Zhou, T., Zhang, Q.M., Stanley, H.E.: The H-index of a network node and its relation to degree and coreness. *Nat. Commun.* **7**(1), Art. No. 1 (2016). <https://doi.org/10.1038/ncomms10168>
10. Liu, Y., Tang, M., Zhou, T., Do, Y.: Identify influential spreaders in complex networks, the role of neighborhood. *Phys. Stat. Mech Appl.* **452**, 289–298 (2016). <https://doi.org/10.1016/j.physa.2016.02.028>
11. Bonacich, P.: Some unique properties of eigenvector centrality. *Soc. Netw.* **29**(4), 555–564 (2007). <https://doi.org/10.1016/j.socnet.2007.04.002>
12. Kim, Y., et al.: Livestock trade network: potential for disease transmission and implications for risk-based surveillance on the island of Mayotte. *Sci. Rep.* **8**(1), 11550 (2018). <https://doi.org/10.1038/s41598-018-29999-y>

13. Ioffe, S.: Improved consistent sampling, weighted Minhash and L1 sketching. In: 2010 IEEE International Conference on Data Mining, Sydney, Australia: IEEE, pp. 246–255 (2010). <https://doi.org/10.1109/ICDM.2010.80>
14. Yeung, K.Y., Ruzzo, W.L.: Details of the Adjusted Rand index and Clustering algorithms Supplement to the paper An empirical study on Principal Component Analysis for clustering gene expression data (to appear in Bioinformatics)
15. Ciss, M., et al.: Description of the cattle and small ruminants trade network in Senegal and implication for the surveillance of animal diseases
16. Reimann, C., Filzmoser, P., Garrett, R.G.: Background and threshold: critical comparison of methods of determination. *Sci. Total Environ.* **346**(1), 1-16 (2005). <https://doi.org/10.1016/j.scitotenv.2004.11.023>
17. Belgiu, M., Drăguț, L.: Random Forest in remote sensing: A review of applications and future directions. *ISPRS J. Photogramm. Remote Sens.* **114**, 24-31 (2016). <https://doi.org/10.1016/j.isprsjprs.2016.01.011>
18. Apolloni, A., et al. :Towards the description of livestock mobility in Sahelian Africa: some results from a survey in Mauritania. *Plos One* **13**(1), e0191565 (2018). <https://doi.org/10.1371/journal.pone.0191565>
19. Wright, E.A.P., Yoon, S., Ferreira, A.L., Mendes, J.F.F., Goltsev, A.V.: The central role of peripheral nodes in directed network dynamics. *Sci. Rep.* **9**(1), Art no. 1 (2019). <https://doi.org/10.1038/s41598-019-49537-8>
20. Nigeria | History, Population, Flag, Map, Languages, Capital, & Facts | Britannica. <https://www.britannica.com/place/Nigeria> (consulté le 17 août 2023)
21. Nicolas, G., Apolloni, A., Coste, C., Wint, G.R.W., Lancelot, R., Gilbert, M.: Predictive gravity models of livestock mobility in Mauritania: the effects of supply, demand and cultural factors. *Plos One* **13**(7), e0199547 (2018). <https://doi.org/10.1371/journal.pone.0199547>
22. Colman, E., Holme, P., Sayama, H., Gershenson, C.: Efficient sentinel surveillance strategies for preventing epidemics on networks. *Plos Comput. Biol.* **15**(11), e1007517 (2019). <https://doi.org/10.1371/journal.pcbi.1007517>
23. Bai, Y., Yang, B., Lin, L., Herrera, J.L., Du, Z., Holme, P.: Optimizing sentinel surveillance in temporal network epidemiology. *Sci. Rep.* **7**(1), 4804 (2017). <https://doi.org/10.1038/s41598-017-03868-6>



Travel Demand Models for Micro-Level Contact Network Modeling

Diaoulé Diallo^(✉), Jurij Schönfeld, and Tobias Hecking

German Aerospace Center (DLR) Linder Höhe, 51147 Cologne, Germany
diaoule.diallo@dlr.de

Abstract. In the pursuit of accurate infectious disease forecasting, micro-level contact modeling in contact networks emerges as a pivotal element. This research delves into the intricacies of nuanced micro-level modeling, presenting adaptable models tailored for specific locations, derived from a refined travel demand model. In our experiments, we observed that varied encounter patterns among individuals directly influence infection dynamics. Additionally, we observe distinct trends in the spreading dynamics between temporal dynamic networks and their static counterparts for certain encounter models. The study underscores the need for a deeper appreciation of micro-level encounter patterns in epidemiological modeling. Such understanding is pivotal in shaping effective interventions and public health strategies during pandemic scenarios.

Keywords: contact networks · micro-level encounter modeling · mobility data · pandemic research · temporal networks

1 Introduction

Mobility is fundamental to epidemic research, as it leads to the formation of complex contact networks through people's spatial encounters at various locations. Such contact networks offer insights into epidemic dynamics and therefore mitigation strategies, and public health policies [1–3, 5].

The significance of contact networks became particularly evident during the global COVID-19 pandemic, where researchers focused intensely on leveraging these networks to both characterize and forecast the spread of the virus [4, 15, 20]. The essence of interactions was elegantly encapsulated within these networks, providing a macroscopic view of transmission dynamics. It is noteworthy that these networks predominantly operated at a macro scale, often dealing with high-level representations, such as compartments or similar abstractions.

However, mobility data on a microscopic level are hard to obtain especially because of technical difficulties and privacy reasons. Recognizing the need for agile responses, we put forth a range of resource-efficient, adaptable, and parameterizable methods designed for modeling individual encounters within distinct locations as temporal dynamic networks.

We explore techniques for micro-level contact modeling with agent-based simulation based on traffic demand models and conceptualize the underlying techniques. A micro-level contact denotes the actual physical encounter between individuals. In this context, we deploy temporal dynamic networks, grounded by various encounter models and distinct types of locations derived from a travel demand model. Precisely capturing low-level encounters forms a valuable part that seamlessly integrates into comprehensive models, augmenting the precision of epidemic forecasting and characterization. Beyond the global context, this research also opens up promising avenues for individual infection risk approximation. The incorporation of location-induced variations in infection risk into digital contact tracing strategies holds substantial potential, advancing the way we approach contact tracing and containment efforts. In the future, travel demand models may serve as a foundational resource for rapidly generating temporal dynamic networks, enabling their versatile application in pandemic response and related endeavors.

The subsequent sections of this paper unfold as follows: Sect. 2 sheds light on methodologies employed in micro-level encounter modeling. Following this, the methodology section initially explores temporal dynamic networks, then proceeds to introduce general distinct approaches to modeling micro-level contacts. The results section compares the outcomes of the three techniques we have employed. Finally, the conclusion section encapsulates the overarching findings and implications that emerge from our study.

2 Background

Two large pandemic simulation models OpenABM [9] and Covasim [11] use the concept of multi-layer networks to generate contact networks for different daily life scenarios (school, work, household, ...). Both models were used to investigate COVID-19 dynamics and test different intervention strategies. The multi-layer network approach makes use of census data to build a synthetic population on an urban scale. Contacts are generated by different models representing different types of interactions and environments in daily life. Covasim generates fully connected networks within households, small world networks on the community and work level, and disconnected clique networks representing classes. Similarly, OpenABM employs fully connected networks at the household level, random networks for communities, and small-world networks for occupations. Both models understand the necessity for different micro-level approaches in different locations. However, they choose different and quite simplistic approaches in the same scenarios, proving the need for further research on that topic.

A study conducted by [12] harnessed mobility data to construct micro-level person encounters. This approach considers temporal intersections of individuals at locations, as well as the type of location. The authors differentiate three location types with each being associated with three basic transmission probabilities. A final edge transmission weight is computed by combining the location-dependant transmission risk and a score derived from the intersection time of two

individuals. However, for any given location, the edge weights are solely dependent on intersection times, overlooking the spatial attributes of the location and individual movement patterns.

Müller et al. [16] used mobile phone data for agent-based epidemiological simulations including factors like masks and air exchange rates. To model micro-level contact encounters, the approach divides locations into subspaces of predetermined capacity, giving rise to a contact network characterized by cliques. While this leads to a sophisticated model for location-based person-to-person encounters, it requires access to mobile phone data and does not fully account for the diverse encounter patterns that different location types exhibit.

The dynamics of disease spreading in various indoor environments has also been explored by several studies using sophisticated simulation techniques [7, 13, 17, 22]. Notably, these investigations have aimed to provide insights into transmission patterns and infection potentials in specific settings where a high amount of information is available. However, the effectiveness of such approaches relies on available and accurate information, e.g. layout, structure, and architecture of the location under investigation, which limits its applicability to settings with varying spatial configurations.

Up until now, the landscape of micro-level contact modeling has been characterized by two predominant trends: network generators that mainly rely on time spent at locations as well as the associated capacities and complex physical simulations necessitating substantial data and computational resources for agent-based modeling. While the former overlooks important interaction dynamics, the latter is resource and data-intensive and may not be feasible in many scenarios.

In the following section, we outline approaches for capturing location-specific encounter patterns based on traffic-demand models without the need for physical simulations.

3 Methodology

In this section, we detail the methodologies foundational to our exploration of micro-level encounter modeling using temporal-dynamic networks. We begin by illuminating the essence of temporal-dynamic networks. Subsequently, we introduce three distinct approaches for micro-level contact modeling.

3.1 Temporal-Dynamic Contact Networks

Temporal-dynamic networks serve as a sophisticated framework that reveals the ever-changing nature of interactions among individuals [10]. In contrast to static networks, which offer a snapshot of connections, temporal-dynamic networks capture the intricate evolution of relationships over time. This real-time depiction introduces a higher level of realism, as interactions are not treated as fixed entities but rather as dynamic occurrences. Temporal-dynamic networks prove invaluable in epidemiological studies, as they grant insights into the spread of

diseases over time [14]. By incorporating time-varying edges, these networks portray the varying transmission potentials at different stages of an epidemic. This precision empowers researchers and policymakers to devise strategies for disease containment and control more effectively.

Temporal-dynamic networks present interactions as evolving sequences, not mere snapshots [19]. While dynamic networks are gaining traction in pandemic research, many studies still rely on static networks due to their computational simplicity. Although static networks can suffice when disease dynamics align with network changes, they can introduce biases. Such biases arise when aggregating variable dynamic contacts, leading to misrepresentations in potential infection paths. It is debated that static networks might intensify infection dynamics. Contrary some cases are known where temporal correlations accelerate the dynamics of stochastic processes in dynamic networks compared to their static equivalent. In [18], SIR simulations were performed on an empirical temporal network of sexual interactions, to investigate the spreading of sexually transmitted infections. Their findings suggest that especially in the early pandemic stage, temporal correlations in the network accelerate infection dynamics leading to higher outbreak sizes, compared to different variations of static network representations. For a deeper understanding, our study examines both dynamic networks and their static counterparts.

In this study, we generate temporal-dynamic micro-level contact networks from mobility data, typically presented as a collection of trajectories that depict the movement of nodes between various locations. Within the realm of contact networks, our focus is on trajectories that conclude at a specific location. We also consider the time elapsed until the subsequent trajectory relocates our node to another location. Based on this, we construct the vector

$$V(t) = (v_1(t), v_2(t), \dots, v_i(t), \dots, v_{N_V}(t))^T,$$

representing all nodes, where $v_i(t) = 1$ if node i is present at our location at time t , and $v_i(t) = 0$ otherwise. Here, N_V denotes the total number of nodes.

In the most general description of a micro-level contact model

$$\theta : V(t) \rightarrow A(t),$$

one takes $V(t)$, which is modulated by mobility data, and uses the contact network model θ to generate the edges of the dynamic network. $A(t)$ represents the time-dependent adjacency matrix, where $a_{i,j}(t)$ is set to 1 if nodes i and j are connected at time t . The equivalent static network consists only of a single adjacency matrix, where $a_{i,j}$ holds the time fraction nodes i, j where in contact during the day.

In the following, possible realizations of the model θ are introduced. These are based on previous work on micro-level contact networks discussed in Sect. 2, and adapted to data resulting from travel demand models.

3.2 Mobility Data

To generate mobility data, we employ the TAPAS (Travel and Activity Patterns Simulation) travel demand model [6]. TAPAS offers a comprehensive framework for simulating future transportation demand scenarios. It takes into account various factors such as demographic changes, income structures, and transportation infrastructure.

The foundation for TAPAS is empirical spatial and structural data, details related to how individuals allocate their time, and specifics regarding transportation mode preferences. This data is drawn from the “Mobility in Germany” (MiD)¹ survey, which collected detailed information from over 316,000 individuals across 156,000 households. This data encompasses activity types, durations, socio-demographic attributes, and household transportation resources. The synthesized data produced by TAPAS results in 24h of data. At each location, individuals are identified along with their arrival and departure times. For our experimental purposes, we selected four specific locations. Two of these locations, A and B, are associated with leisure activities, while location C represents a workplace and location D stands for a school.

To conduct a comprehensive SIR simulation across several days, we address the challenge posed by the availability of accurate mobility data for just one day. Our approach involves stacking the temporal contact network data from this single day to simulate a continuous span of 20 days. While this method doesn’t fully capture the stronger fluctuations and long-term spreading potentials that may emerge among communities and individuals over time, it serves our primary purpose effectively. Our main interest lies in uncovering general topological differences exposed by SIR simulations across various modeling approaches. By extending the available data in this manner, we can gain valuable insights into the impact of micro-level encounter modeling on the broader epidemic dynamics.

3.3 Micro-Level Contact Modeling

Baseline Approach $\theta_{baseline}$: Our baseline approach builds upon the work of Klise et al. [12]. In essence, this method leverages mobility data and individual-specific time allocations at specific locations to compute intersecting time frames between individuals, subsequently constructing contact networks.

In this approach, individuals present at the same location are linked by edges in a contact network, with edge weights determined by the shared duration of their presence. Transforming this concept into a temporal dynamic network, we establish edges connecting pairs of individuals who coincide at a given point in time within the same location (see 3.1). Under this premise, our approach assumes an equal likelihood of infection for any pair of individuals who share the same duration of stay at a location. In other words $\theta_{baseline}$ constructs a fully connected network between all nodes active at time t . This simplified framework

¹ <https://bmdv.bund.de/EN/Services/Statistics/Mobility-in-Germany/mobility-in-germany.html>.

forms the foundation of our exploration, serving as a reference point against which we compare our more intricate modeling techniques.

Random Graph-Based Approach θ_{random} : In our random graph-based approach, similar to [9], every possible edge, meaning that node i, j are present at the location at time t , is selected with probability p_{random} . Additionally, a contact duration is drawn from an exponential distribution with mean β . Contacts, therefore, have a minimum duration of one time step and, in the case of non-consecutive contacts, a mean duration of $1 + \beta$ time steps. This distribution accounts for the variable nature of interaction durations, resulting in a dynamic and realistic representation of human encounters. A possible application would be in locations where interactions are mainly random and short, like in supermarkets, where the case of two individuals being in close proximity for the entire shopping trip is rather unlikely, however frequent but short contacts are to be expected.

Clique-Based Approach θ_{clique} : This approach capitalizes on the concept of forming cliques to model micro-level encounters, advancing the clique-based strategy of [16]. By grouping individuals into these compact clusters, we create an efficient representation of contact networks within specific environments. This approach is particularly useful for capturing interactions in places with constrained capacity, like offices or classrooms. First, individuals are assigned to spaces within the location, with fixed size $N_{PeoplePerSpace}$. Nodes enter the location and their respective space according to $V(t)$, forming tightly bounded cliques. For contacts between different spaces at every time step, a node changes its space with probability p_{clique} for a duration that is drawn from a normal distribution $\mathcal{N}(\mu, \sigma)$. Afterwards, the node goes back to its default space.

By modeling and tracing movements within these spaces over time, we identify instances of shared occupancy. These instances lead to the formation of cliques, where individuals have pronounced edges connecting them within the clique, reflecting intensive interactions like in shared offices or classrooms. In contrast, connections outside the clique are rare, mirroring more sporadic or distant interactions. The underlying idea of this approach is to encapsulate the nuanced interplay between spatial arrangements and interpersonal encounters. This modeling technique ensures a more comprehensive understanding of how individuals' interactions are influenced by their physical proximity within specific locations.

3.4 Unveiling Topological Properties with SIR Model

To assess the topological differences introduced by our various micro-level contact network modeling approaches, we employ the Susceptible-Infectious-Recovered (SIR) model [8]. The SIR model is a well-established compartmental model used to analyze the spread of infectious diseases within a population. It divides individuals into three compartments: susceptible (S), infectious (I), and recovered (R). The SIR model tracks the transitions of individuals between these compartments based on their interactions and the disease's transmission dynamics. For our evaluation, we utilize a temporal dynamic SIR model implemented

using the Tacoma framework². Tacoma provides a versatile platform for studying epidemic spreading and other dynamical processes on networks utilizing the Gillespie algorithm [21]. We let the epidemic spreading simulations run for a simulated period of 20 artificial days. During this time, we monitor the progression of the infection within the population and observe how different modeling approaches influence the spread of the disease. This SIR-based evaluation allows us to gain insights into the impact of micro-level encounter modeling on the topological properties of contact networks and the resulting epidemic dynamics. By analyzing the simulated disease propagation under different scenarios, we can draw conclusions about the importance of accurately representing individual interactions for understanding and managing the spread of infections.

In our research, it is crucial to recognize that the distinctive nature of our various approaches inherently results in networks from the same location having varied edge counts but identical node counts. The baseline approach, as described in Sect. 3.3, exhibits a markedly higher mean degree. To ensure a valid and unbiased comparison using a SIR model, we assume that the interaction strength is constant across all networks, i.e.

$$\sum_{t,i,j \in E_{baseline}} w_{i,j}(t) = \sum_{t,i,j \in E_{random}} w_{i,j}(t) = \sum_{t,i,j \in E_{clique}} w_{i,j}(t) = 1$$

where E is the respective set of edges generated by the contact network model. This is achieved by normalizing the adjacency matrix $w_{i,j}(t) = \frac{a_{i,j}(t)}{N_E}$ with the total number of edges N_E during the day creating weighted edges. The experimentation involved adjusting the transmission probability parameter to show sufficient infection dynamics across all networks. SIR runs were performed with this transmission probability and with respect to the edge weights. This methodology guarantees a meaningful assessment of the impact of different network topologies on the dynamics of disease propagation, even when the networks exhibit varying edge counts by definition.

4 Results

In this section, we present our results on micro-level contact network modeling using mobility data. We first describe the mobility data utilized and then discuss our experimental results. For all experiments, we selected $N_{PeoplePerSpace} = 15$, $p_{clique} = 0.01$, $\mu = 10$, $\sigma = 5$. These parameter choices were informed by preliminary experiments and explored in Sect. 4.2.

4.1 SIR-Based Evaluation

The outcomes of the SIR simulation conducted over a span of 20 days are presented in Fig. 1. The vertical axis on the graph represents the number of individuals infected per day. Evidently, the baseline approach exhibited the highest

² <https://github.com/benmaier/tacoma>.

infection count, followed sequentially by the random and clique approaches. This consistent trend was observed across all examined locations. Importantly, the distinctions among the approaches go beyond just the highest infection count, also encompassing differences in the rate of spread. For both, locations A and B, it becomes evident that the peak of the clique approach occurred around days 7–8, whereas the baseline approach reached its highest point at approximately day 5. The work/school locations C and D show similar trends but the difference in infections and speed between the approaches is less emphasized.

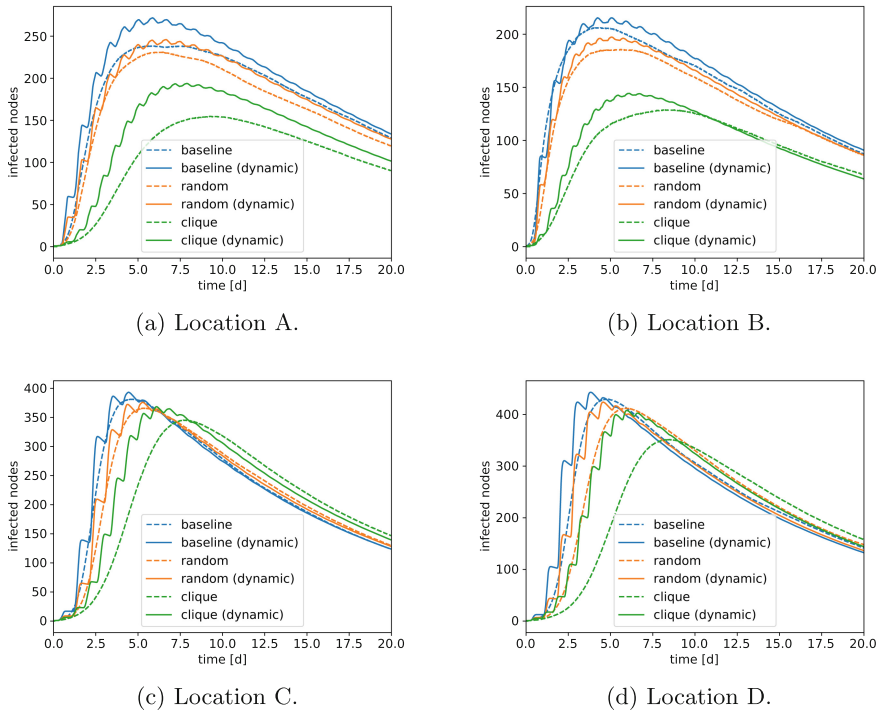


Fig. 1. Comparison of infection dynamics in SIR simulation across multiple locations and contact modeling approaches. Number of infected nodes on the y-axis, number of days on the x-axis. $N_{PeoplePerSpace} = 15$, $p_{clique} = 0.01$, $\mu = 10$, $\sigma = 5$.

Essentially, we find variations in the extent and speed of infection dynamics across the selected locations, which can be tied to the nature of each location. For instance, individuals tend to spend less time at locations A and B, which are associated with leisure activities, compared to those representing a school or a workplace. Crucially, the distinct contact models shaping encounter patterns play a significant role in influencing the spreading dynamics within the constructed contact networks.

The difference between dynamic and their respective static networks depends on the chosen model. For our baseline, as well as the random approach, we see that both networks behave similarly in terms of infection dynamics. The clique approach instead shows a significant acceleration of the infection dynamic in dynamic networks, resulting in a higher and earlier infection peak as well as a higher outbreak size. This further supports the findings from [18]. Their sexual encounter network is described as a network with many cycles and compact weekly connected cliques, resulting from spatial constraints. This network could be modeled with our clique approach.

The actual efficacy and adaptability of these models necessitate further validation, either through empirical data or simulation studies. Nevertheless, these models serve as starting point and aim to lay the groundwork for fast and adaptable generation of micro-level contact models.

4.2 Effect of Hyperparameter Settings

Figure 2 demonstrates the effects of various hyperparameters associated with the clique-based approach on the SIR results for location B. Except for the parameter under investigation, we maintain consistency with the experiments detailed in Sect. 4.1. Figure 2a reveals the influence of the mean parameter μ . Elevated μ values correlate with a rise in total infections and a decelerated infection spread. The same trend surfaces when observing the number of individuals per space, $N_{PeoplePerSpace}$. Infections peak around 50 for $N_{PeoplePerSpace} = 5$ and approximately 170 for $N_{PeoplePerSpace} = 50$. Likewise, Fig. 2b shows that alterations in the probability of space change p_{clique} lead to varying infection outcomes. As expected, larger probability values result in higher infection counts, while minimal space changes yield minimal infections. The absence of space changes restricts inter-clique infections, resulting in substantially lower infection dynamics. Conversely, fluctuations in the σ parameter exhibit minimal impact on infection dynamics, as Fig. 2d shows. Since this parameter influences the time individuals spend in other spaces without directly affecting encounter numbers, it appears to play a less pronounced role in driving infection dynamics. Upon inspecting the network resulting from θ_{random} , the hyperparameters showcased minimal perturbation on the outcomes. Our experiments revealed negligible effects for β and only minor variations observed in the infection dynamics for p_{random} . When exploring values for p_{random} spanning from 0.001 to 0.5, a notable reduction in infection rate was evident at 0.001, while the remaining values demonstrated relatively comparable results. Notably, $p_{random} = 0.02$ exhibited the highest infection rate, closely followed by 0.3. Neither the contact duration nor the number of edges (under the normalizing factor) appeared to exert a significant impact on the network's topology.

In essence, the results indicate that constrained spaces and diminished interactions among occupants lead to reduced infection propagation. This observation is independent of the sheer edge count since the transmission probability between nodes is normalized as explained above. The deviation in outcomes stems from the unique topological traits of the temporal contact network, reflecting varying

encounter dynamics. Our research accentuates the pivotal role of assumptions surrounding encounter patterns and consequent transmission dynamics in dictating infection trajectories. Utilizing contact networks constructed from comprehensive data sources, like mobility data, reveals the importance of real-world contact patterns in epidemiological modeling.

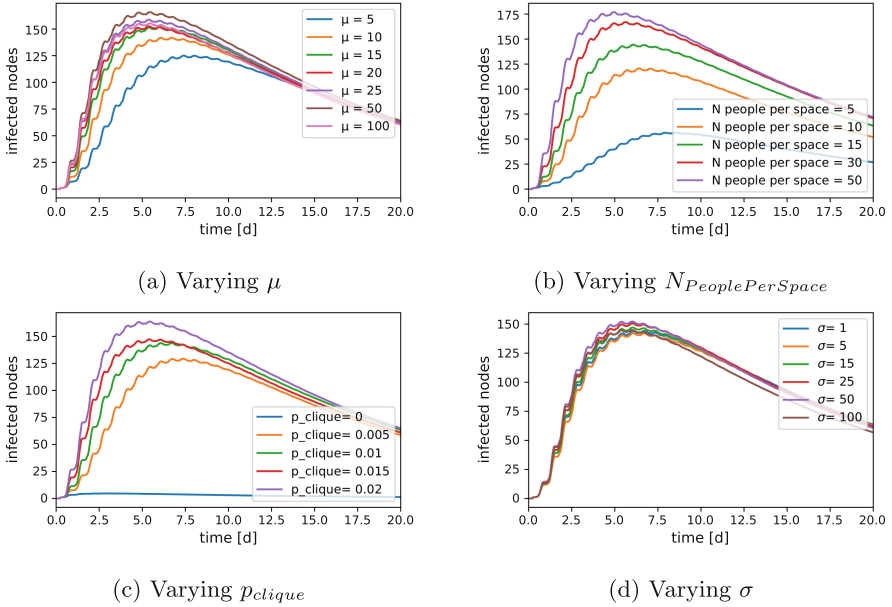


Fig. 2. Parameter exploration for temporal contact network resulting from θ_{clique}

While our study has provided insights into the behavior and characteristics of temporal contact networks, limitations need to be acknowledged. Our current method of stacking these networks doesn't capture long-term dynamics of infection spread. While our choice of an SIR-based evaluation provides a foundation, the process of normalizing temporal networks introduces complexities, as the "overall infection potential" is differently interpreted. Distinct characteristics between temporal and static networks are subject to future investigations and underpin the relevance of temporal dynamic network modeling. While our approach offers promising avenues for future research, its broader applicability needs cautious consideration and further refinement.

5 Conclusion

In this study, we explored micro-level contact network modeling and its implications for understanding disease spread. The goal was to showcase how the

choice of micro-level contact models within specific locations influences infection dynamics. Our findings highlight the significance of tailored contact models for different locations and the crucial role of encounter patterns in shaping infection dynamics. Employing travel demand models in understanding infection dynamics paves the way for flexible and modifiable contact models. This research underscores the complexity of real-world contact patterns in epidemiological modeling, emphasizing the need for nuanced approaches to inform public health strategies. The orchestration of multi-tiered contact networks necessitates authentic portrayals of human mobility on both macro and micro scales, enriching our competence in offering precise infection risk assessments to individuals.

Future research directions include refining transmission probability modeling by considering contact distance through human mobility models that emulate actual human movement patterns. Furthermore, analyzing long-term data spanning more than 24 h can reveal longer-term effects and pave the way for the generation of even more accurate temporal contact networks. Additionally, we aim to develop a versatile, parameterizable model applicable to various location types to enhance its adaptability and usefulness in epidemiological investigations.






References

- Balcan, D., Gonçalves, B., Hao, H., Ramasco, J.J., Colizza, V., Vespignani, A.: Modeling the spatial spread of infectious diseases: the GLocal epidemic and mobility computational model. *J. Comput. Sci.* **1**(3), 132–145 (2010)
- Dalziel, B.D., Pourbohloul, B., Ellner, S.P.: Human mobility patterns predict divergent epidemic dynamics among cities. *Proc. Roy. Soc. B: Biol. Sci.* **280**(1766), 20130763 (2013)
- Eubank, S., et al.: Modelling disease outbreaks in realistic urban social networks. *Nature* **429**(6988), 180–184 (2004)
- Firth, J.A., Hellewell, J., Klepac, P., Kissler, S., Kucharski, A.J., Spurgin, L.G.: Using a real-world network to model localized COVID-19 control strategies. *Nat. Med.* **26**(10), 1616–1622 (2020)
- Glass, L.M., Glass, R.J.: Social contact networks for the spread of pandemic influenza in children and teenagers. *BMC Public Health* **8**(1), 61 (2008)
- Heinrichs, M.: TAPAS: travel-activity-pattern simulation - parallelisiertes mikroskopisches verkehrsnachfragemodell. In: Next GEneration Forum 2011, pp. 74–74. Deutsches Zentrum für Luft und Raumfahrt e.V. (2011)
- Hekmati, A., Luhar, M., Krishnamachari, B., Matarić, M.: Simulating COVID-19 classroom transmission on a university campus. *Proc. Natl. Acad. Sci.* **119**(22), e2116165119 (2022)
- Hethcote, H.W.: The mathematics of infectious diseases. *SIAM Rev.* **42**(4), 599–653 (2000)
- Hinch, R., et al.: OpenABM-covid19-an agent-based model for non-pharmaceutical interventions against COVID-19 including contact tracing. *PLoS Comput. Biol.* **17**(7), e1009146 (2021)
- Holme, P., Saramäki, J.: Temporal networks. *Phys. Rep.* **519**(3), 97–125 (2012)
- Kerr, C.C., et al.: Covasim: an agent-based model of COVID-19 dynamics and interventions. *PLoS Comput. Biol.* **17**(7), e1009149 (2021)

12. Klise, K., Beyeler, W., Finley, P., Makvandi, M.: Analysis of mobility data to build contact networks for COVID-19. *PLoS ONE* **16**(4), e0249726 (2021)
13. Lee, B., et al.: Designing a multi-agent occupant simulation system to support facility planning and analysis for COVID-19. In: *Designing Interactive Systems Conference 2021*, pp. 15–30. ACM (2021)
14. Leitch, J., Alexander, K.A., Sengupta, S.: Toward epidemic thresholds on temporal networks: a review and open questions. *Appl. Network Sci.* **4**(1), 105 (2019)
15. Liu, F., Li, X., Zhu, G.: Using the contact network model and metropolis-Hastings sampling to reconstruct the COVID-19 spread on the “diamond princess.”. *Sci. Bull.* **65**(15), 1297–1305 (2020)
16. Müller, S.A., et al.: Predicting the effects of COVID-19 related interventions in urban settings by combining activity-based modelling, agent-based simulation, and mobile phone data. *PLoS ONE* **16**(10), e0259037 (2021)
17. Reveil, M., Chen, Y.-H.: Predicting and preventing COVID-19 outbreaks in indoor environments: an agent-based modeling study. *Sci. Rep.* **12**(1), 16076 (2022)
18. Rocha, L.E.C., Liljeros, F., Holme, P.: Simulated epidemics in an empirical spatiotemporal network of 50,185 sexual contacts. *PLoS Comput. Biol.* **7**(3), 1–9 (2011)
19. Sharkey, K.J., et al.: Pair-level approximations to the SPATIO-temporal dynamics of epidemics on asymmetric contact networks. *J. Math. Biol.* **53**(1), 61–85 (2006)
20. Thurner, S., Klimek, P., Hanel, R.: A network-based explanation of why most COVID-19 infection curves are linear. *Proc. Natl. Acad. Sci.* **117**(37), 22684–22689 (2020)
21. Vestergaard, C.L., Génois, M.: Temporal Gillespie algorithm: fast simulation of contagion processes on time-varying networks. *PLoS Comput. Biol.* **11**(10), 1–28 (2015)
22. Ying, F., O’Clery, N.: Modelling COVID-19 transmission in supermarkets using an agent-based model. *PLoS ONE* **16**(4), e0249821 (2021)



Evaluating Attitudes on Health-Seeking Behavior Among a Network of People Who Inject Drugs

Ayako Shimada¹ , Ashley L. Buchanan² , Natallia V. Katenka³  , Benjamin Skov², Gabrielle Lemire², Stephen Kogut² , and Samuel R. Friedman⁴

¹ Division of Biostatistics, Department of Pharmacology and Experimental Therapeutics, Thomas Jefferson University, Philadelphia, PA 19107, USA

² Department of Pharmacy Practice, College of Pharmacy, University of Rhode Island, Kingston, RI 02881, USA

³ Department of Computer Science and Statistics, College of Arts and Sciences, University of Rhode Island, Kingston, RI 02881, USA
nkatenska@uri.edu

⁴ Department of Population Health, New York University, Grossman School of Medicine, New York University, New York, USA

Abstract. People who inject drugs (PWID) are part of HIV/AIDS risk networks, where individuals can engage in sexual and injection risk behavior. Low socioeconomic status and lack of access to medical care often complicate successful engagement in HIV care for PWID. This study investigates how locus of control and self-blame regarding HIV/AIDS risk affects health-seeking behavior in PWID participants and their community members. We apply causal inference methodology to PWID HIV risk networks ascertained from the Social Factors and HIV Risk Study (SFHR) conducted between 1991 and 1993 in Bushwick, Brooklyn, New York. We found estimated protective disseminated effects of attitudes toward HIV/AIDS on health-seeking behaviors of others in the PWID community. A positive attitude toward controlling HIV/AIDS can improve the health-seeking behavior of other members in the community with a pessimistic attitude toward HIV/AIDS control. Interventions to improve attitudes toward HIV/AIDS risk can boost health-seeking behavior among both PWID receiving the intervention themselves and other unexposed PWID in the community.

Keywords: Causal Inference · Health Attitudes · HIV/AIDS · Injection Drug Use · Risk Network

1 Introduction

HIV/AIDS remains a significant concern among people who inject drugs (PWID) in the United States [1]. Sexual and injection behaviors can increase the risk of HIV transmission [2]. Compounding the issue, PWID with low socioeconomic backgrounds often struggle with accessing adequate medical care, hindering successful engagement with

the HIV care [3]. PWID are often part of HIV risk networks, where partnerships are defined by sexual or injection risk behaviors. PWID are face both an increased risk of HIV infection also unique barriers that can limit their engagement with HIV care [3].

One class of *network-targeted interventions* attempts to leverage dissemination in a social network for behavioral change [4–6]. Among a network, dissemination (spillover, interference) can occur when members of a social group modify their behavior based on the traits, beliefs, attitudes, or norms among their social contacts, specifically when one individual's health attitude affects another individual's health outcome. This dissemination of attitudes may be possible through peer influence, for example, due to the verbalization of attitudes or behavior modeling [4, 7]. Although potential HIV transmission networks have dependence structures associated with sexual and injection risk behaviors, this dependence structure is related to but not synonymous with dissemination in networks. Our work focuses on dissemination through HIV risk networks. In earlier work, dissemination was conceptualized as the framework to analyze the dependent events of infectious diseases with four different effects of interest in a two-stage randomized design, where investigators randomly assign a treatment allocation strategy (i.e., vaccination coverage in a community) to each community then assign actual treatment (i.e., vaccine) to participants in each community given the assigned treatment allocation strategy [8, 9]. In recent work, the methods to estimate the four parameters using data from observational studies have been developed [10, 11].

Interventions can be strengthened by assessing dissemination to inform more effective and sustainable solutions for HIV/AIDS prevention among PWID [12, 13]. However, prior to the introduction of a network-based intervention to change attitudes, an initial step is to evaluate the relationship between the attitudes of PWID and their health-seeking behaviors. Although some earlier studies assessed attitudes toward HIV/AIDS risk [14, 15], there are limited studies about the PWID attitudes toward HIV/AIDS risk on their own health-seeking behavior. Evaluating the effects of attitudes toward HIV/AIDS risk on PWID's health-seeking behavior can provide substantially new insights for developing more effective and sustainable interventions. This study quantifies the relative magnitudes of PWID's attitudes toward HIV/AIDS risk on their own health-seeking behaviors and on other individuals in their risk network.

2 Methods

The SFHR study was conducted in Bushwick, New York 1991 to 1993 [6, 16]. Data were collected from street recruited PWID in the Bushwick neighborhood, a low-income area of approximately 100,000 residents with high rates of poverty, injection drug use, and HIV/STI prevalence. The original study enrolled a total of 767 participants and included 3,162 dyadic relationships. Dyadic relationships were defined as one individual reporting that he/she had sex or injected drugs with another individual in the previous 30 days, and each participant named up to 10 contacts [16].

Of the 3,162 dyadic links in the original SFHR study, 2,498 links recorded were between enrolled and non-enrolled individuals [16]. We excluded these non-enrolled individuals from the analysis, including the 2,498 edges between enrolled and non-enrolled individuals. Eighty-two enrolled participants were missing either outcome,

exposure, and/or covariate information, so we excluded these individuals from the analysis. After this exclusion, we had 283 participants with no observed relationships to other participants in the network. These isolated participants were also removed from the analysis. The SFHR PWID network for this analysis included 402 participants (i.e., vertices or nodes) with 403 risk connections (i.e., edges).

We evaluated communities in the SFHR PWID network using a modularity-based community detection approach. A *community* is defined as a group of participants that is densely connected with only sparser connections to other participants outside of the group [17]. In the SFHR PWID network, these communities included PWID who were more highly connected within the same community but had sparser connections to the PWID in other communities. For the modularity-based method, modularity takes large values when there are more connections among some individuals than expected if connections were randomly assigned, suggesting the presence of a nontrivial community structure in a network [17, 18]. As a result, there are more edges among the participants in a community than between communities in the SFHR network.

We consider two separate exposures to assess attitudes toward HIV/AIDS risk: (1) HIV/AIDS locus of control (i.e., *internal* vs. *external*) and (2) blame attributes (*self-blame* vs. *blame others*). The locus of control is defined as the degree to which an individual believes they have control over what will happen or has happened to themselves and can be classified into two different types: *internal* and *external* [19]. Individuals with an internal locus of control (ILOC) attribute the events they experience to factors within their control, while those with an external locus of control (ELOC) attribute events to factors beyond their influence [20, 21]. HIV/AIDS locus of control is defined as an individual belief about how much control one has over its own HIV/AIDS risk. The blame attribute toward HIV/AIDS is defined as an individual blaming themselves (i.e., *self-blame*) or blaming others or society (i.e., *blame others*) for their perceived HIV/AIDS risk.

Participants were asked ten questions about their health beliefs to determine their HIV/AIDS locus of control and blame attribute (Table 1). Questions 4, 5, and 9 capture information about PWID's individual blame, and the remaining questions are about PWID's HIV/AIDS locus of control. The reverse scale was used for negatively phrased items. The responses were originally recorded on a Likert scale (strongly agree, agree, somewhat agree, disagree, strongly disagree, don't know, refused, and not applicable). To create a binary variable to represent locus of control, for each response out of seven, we assigned the value of 1 if a participant reported ILOC, while the value of -1 was assigned if a participant reported ELOC. If the response was neutral (i.e., don't know), we assigned a value of 0. Adding all values assigned to the responses for health belief related questions, we obtained an individual health belief scores ranging from -7 to 7. Then, if one's locus of control score was greater than or equal to three, the participant was assigned as having an ILOC; otherwise, the participant was assigned ELOC. Because traits that consist of both negative and positive aspects can be negatively biased, we selected a positive threshold [22]. A similar procedure was taken to create a binary variable to represent a participant's blame attribute. We obtained blame scores that ranged from -3 to 3 for individual participants. The distribution of blame scores is shown in the right panel of Fig. 1. If a participant's blame score was equal to three, the attribute was

assigned as “self-blame”; otherwise, the attribute was “blame others.” We verified the internal consistency of the individual responses for the locus of control and blame items with Cronbach’s alpha on their original Likert scale.

Table 1. Questions about health beliefs in the Social Factors and HIV Risk (SFHR) Study from Bushwick, New York, 1991–1993. * Item required a reversed scale in calculating the score for health beliefs. ** Item required a reversed scale in calculating the score for blame attitudes.

SFHR Questions about Health Beliefs
Q1. It is my own behavior which determines whether I get AIDS or not
* Q2. No matter what I do, if I’m going to get AIDS, I will get AIDS
Q3. I’m in control of whether or not I get AIDS
** Q4. My family has a lot to do with whether I get AIDS. (Blame)
Q5. If I get AIDS, I’m to blame. (Blame)
* Q6. Getting AIDS is largely a matter of bad luck
* Q7. No matter what I do, I’m likely to get AIDS
Q8. If I take the right actions, I can avoid getting AIDS
** Q9. If I get AIDS it is because of the society we live in. (Blame)
* Q10. No matter what I do, I’m unlikely to get AIDS

The outcome was defined as the receipt of the SFHR HIV test result. The HIV test was conducted as a part of the SFHR study, and the participants could receive the result after their interview. Therefore, there was a temporal ordering between the exposures and the outcome. Receipt of HIV test results for each participant was recorded as a binary variable (i.e., 1 if “Yes” and 0 otherwise).

Participants’ demographics and characteristics were summarized with descriptive statistics (Table 2). We also created a binary variable indicating participants’ knowledge of their HIV/AIDS status before the SFHR study. We assumed that a participant knows their HIV/AIDS status if they were ever told they had HIV/AIDS or if they were tested at least once before SFHR *and* obtained the last test result; otherwise, we assumed the participant did not know their HIV/AIDS status before SFHR. In the final model of the analysis to adjust for confounding, the following variables were included as covariates: medical payment method (some insurance, paid by self, or other), pre-SFHR knowledge of HIV/AIDS status (Yes vs. No), sex (Male vs. Female), race (White vs. Non-White), age (40–65 vs. 19–39), and the interactions of sex and age, and race and medical payment method to allow the model to be more flexible. For the 96 communities of PWID, we computed the observed distributions of the proportion of PWID reporting ILOC/self-blame in a community (i.e., observed coverage).

2.1 Causal Inference Framework under the Presence of Dissemination

Causal inference in this setting requires several identifying assumptions. Two assumptions are related to the structure of dissemination: *partial interference* and *stratified interference*. The partial interference assumption means that individual's locus of control can affect the receipt of HIV testing results of other individuals in the same community but does not affect others outside of that community [9, 10, 23, 24]. The stratified interference assumption means that an individual's potential outcome depends only on his/her own locus of control and also the proportion of those community members reporting ILOC [9, 10]. We also make the following three assumptions: i) conditioning on a vector of pre-exposure covariates (e.g., medical payment method, pre-SFHR knowledge of HIV/AIDS status, sex, race, age, and the interactions of sex and age, and race and medical payment method), the vector of locus of control exposures for a community is independent of community-level potential outcomes (i.e., *community-level exchangeability*), ii) for each pre-exposure covariate, there is a positive probability of each level of the community-level exposure (i.e., *community-level positivity*), and iii) the exposure is well defined, which means an individual either has ILOC or ELOC and if there are other versions of locus of control, we assume that they are irrelevant to the causal effects of interest. We assume there is no misclassification of attitudes towards HIV/AIDS; that is, every participant correctly reports his/her attitudes in the study, and this exposure accurately captures the underlying attitudes. We also assume the model for the exposure weights is correctly specified (e.g., correct functional forms of covariates), and there is no homophily (plausible in the SFHR study because it is unlikely that PWID formed risk connections based on whether they would receive their HIV test results); that is, individuals in the network are not forming HIV risk connections based on some unobserved variables also associated with their health-seeking behaviors [25], and missing outcomes, exposures, or covariates are missing completely at random (MCAR) [26].

We are interested in four different causal effects in the presence of dissemination [9, 10]. In the following, Y_{ij} , A_{ij} , X_{ij} represent the observed outcome of receipt of SFHR HIV testing result, attitude status, and covariate vector of the j^{th} individual in community i , respectively. Also, \mathbf{A}_i and \mathbf{X}_i are vectors of exposures (i.e., locus of control/blame attributes) and covariate matrices for members within community i , respectively. In our setting, the coverage (i.e., allocation strategy) α or α' is defined as the probability of PWID reporting ILOC/self-blame in a community (α , α' where $\alpha < \alpha'$). The average potential outcome $\bar{Y}(a; \alpha)$ of the j^{th} individual in community i (averaged over all possible Bernoulli allocations for the community, then averaged within and across communities) depends on the exposure of that individual and the coverage of exposure of others in community i .

The following notations represent four different causal effects of interest [9]. The population-level estimands and estimators discussed herein pertain specifically to the study population (i.e., PWID in Bushwick, New York) rather than the broader and hypothetical underlying target population. The *direct* effect, which compares population-level average potential outcomes under the exposure ELOC/blame others to ILOC/self-blame under a coverage level α of ILOC, is defined as $DE(\alpha) = \bar{Y}(a = 0; \alpha) - \bar{Y}(a = 1; \alpha)$. The *indirect* (or disseminated) effect, which compares population-level average potential outcomes with ELOC/blame others under different coverage levels of ILOC is defined

as $\overline{IE}(\alpha, \alpha') = \overline{Y}(a = 0; \alpha) - \overline{Y}(a = 0; \alpha')$. The *total* (or composite) effect, which is the sum of direct and indirect effects and can be interpreted as the maximal impact of exposure to attitudes at population-level, is defined as $\overline{TE}(\alpha, \alpha') = \overline{Y}(a = 0; \alpha) - \overline{Y}(a = 1; \alpha')$. The *overall* effect, which is interpreted as a comparison between communities with different coverage levels of ILOC is: $\overline{OE}(\alpha, \alpha') = \overline{Y}(\alpha) - \overline{Y}(\alpha')$. Then, population-level inverse probability weighted (IPW) estimator of the direct effect is represented by: $\widehat{DE}(\alpha) = \widehat{Y}^{ipw}(a = 0; \alpha) - \widehat{Y}^{ipw}(a = 1; \alpha)$, where \widehat{Y}^{ipw} is the average of community-level average estimated potential outcomes. The remaining population-level IPW estimators of the indirect, total, and overall causal effects are defined analogously. When quantifying these parameters in an observational study, community-level propensity scores are estimated using the information of individual-level covariates in a mixed effects logit model with a random effect for correlation in each community. The inverse of the estimated propensity score is used as an exposure weight in the estimator of interest (i.e., a contrast of estimated average community-level potential outcomes) [10]. We used robust variance estimators accounting for weights estimation to construct 95% Wald-type confidence interval (CIs) for the effects [27].

We used SAS 9.4 for data preparation and R version 3.4.4 for the visualization and analysis. For the estimation in the presence of interference, we used “inference” package in R (<https://cran.r-project.org/package=inference>) [23], which implements the IPW estimation method for observational studies with dissemination [10, 27].

3 Results

Among the 402 participants, the mean age was 35 years and about 70% were male. Most participants had less than a 12th-grade education (64%), were unemployed (91%), and had some insurance for medical expenses (65%). Based on the history of HIV testing, the receipt of the last test result, and knowledge of their HIV/AIDS status, 47% knew their own HIV/AIDS status before the SFHR study (Table 2). In addition, 41% were HIV infected, 5% had developed AIDS, and 75% were positive for Hepatitis B (HB) (Table 2). Of 402 participants, 19% picked up their SFHR HIV testing results.

There were 85 connected components, and one of them formed a giant component, including 199 participants and 275 risk connections (Fig. 1). Using a modularity-based community detection approach, we found 12 communities in the giant component and defined a total of 96 communities in the observed network. Among all 403 risk connections, only 56 were between communities (14%), and 347 were within communities (86%). The average number of participants in a community was 4.2 participants (ranging from 2 to 35 participants). The distribution of coverage of HIV/AIDS self-blame attribute had wider variation than that of ILOC. To ensure enough communities at each coverage level and a range of coverage levels representing moderate to high coverage, we focused our analysis on coverages of 50%, 70% and 99%. The relations between attitudes and the outcome in the SFHR PWID network are visualized in Fig. 1. Reporting ILOC was associated with 87% higher odds of receipt of HIV test results (95% CI: 0.85, 4.11).

Results for both unadjusted and adjusted estimates are displayed in Table 3. The IPW estimates of the population-level causal effects of ELOC on receipt of SFHR HIV test results are displayed in Table 3. Adjusted estimates, i.e., those that controlled for baseline covariates, were comparable to unadjusted estimates but had slightly narrower confidence intervals. For the direct effect estimates, community with 50% and 70% coverages had estimates of similar magnitude for locus of control on the receipt of HIV testing result. In other words, in a community with 50% coverage of ILOC, we would expect 13 more participants to receive their HIV test result per 100 individuals under ILOC exposure compared to ELOC.

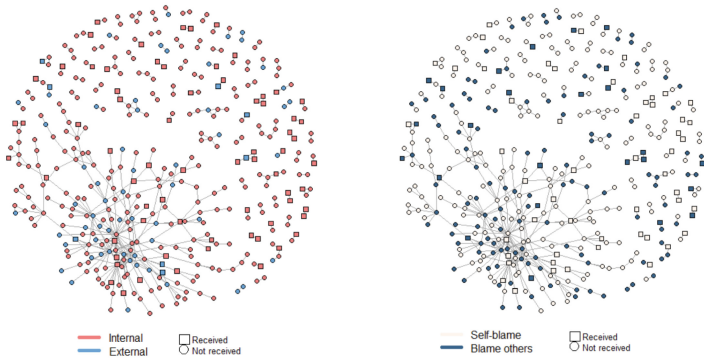


Fig. 1. The Social Factors and HIV Risk Study (SFHR) PWID network for the analysis with 402 nodes and 403 links. The term “Received” indicates the act of a participant picking up the SFHR HIV test result, while “Not received” denotes the absence of such an action.

Interestingly, in community with the highest coverage of 99%, the estimated direct effect was the smallest among the three coverage groups: $\widehat{DE}(99) = -0.10$ (95% CI: -0.26, 0.05). A significant indirect effect was estimated comparing 50% and 70% ILOC coverage communities: $\widehat{IE}(50, 70) = -0.03$ (95% CI: -0.06, -0.01). That is, we would expect 3 more individuals with ELOC to receive their HIV test result in a 70% ILOC coverage community compared to a community with only 50% ILOC coverage. The largest total effect estimate was for the comparison 50% and 70% ILOC coverage communities, and 50% and 99% ILOC coverage communities: $\widehat{TE}(50, 70) = \widehat{TE}(50, 99) = -0.17$ (95% CIs: $-0.27, -0.08$ for 50% versus 70%; $-0.28, -0.07$ for 50% versus 99%). In other words, we expect 17 more participants to receive their HIV test result per 100 individuals if they reported ILOC in 70% ILOC coverage communities compared to individuals who reported ELOC in a 50% ILOC coverage community. Finally, the overall effect estimates indicate that the likelihood of receipt of HIV testing results increases as community coverage increases. For example, we expect that 11 more individuals will receive their HIV test results if a community has 99% ILOC coverage compared to a community with only 50% coverage (95% CI: 0.03, 0.18). None of the estimates for the causal effect of blame on probability of receiving HIV test results were statistically significant. However, Cronbach’s alpha for the blame attribute was quite low (0.77 for belief, and 0.47 for blame) indicating a possible lack of internal consistency for this

Table 2. Participants' characteristics in the Social Factors and HIV Risk (SFHR) Study from Bushwick, New York, 1991–1993 (n = 402).

Characteristics ^a	Number of Participants (%)
Age, Mean (SD)	35 (6.9)
Young Adult (19–39 years old)	290 (72)
Middle Aged (40–65 years old)	112 (27)
Sex	
Male	287 (71)
Female	115 (29)
Race/ethnicity	
White	153 (38)
Other ^b	249 (62)
Highest education	
Less than high school graduation	258 (64)
High school or more	143 (36)
Work status	
No job	364 (91)
Some work	37 (9)
Where currently live	
In your own apartment or house	116 (29)
Someone else's apartment or house	192 (48)
Homeless/other	94 (23)
Medical expense payment method ^c	83 (21)
Pay myself	
Some insurance	262 (65)
Other	57 (14)
Ever told that you have AIDS/HIV	27 (7)
Ever tested for the HIV	234 (58)
Number of HIV tests ever taken, Mean (SD)	1.9 (1.2)
Pick up your HIV test results last time ^d	164 (70)
HIV/AIDS status known before SFHR study ^e	188 (47)
HIV positive	162 (41)
HB core antibody positive	244 (75)
AIDS	20 (5)

^aThere were missing observations in highest education and work status. In the HIV positive, HB core antibody positive, and AIDS variables, there were 6, 77, and 7 missing data, respectively.

^bOther category included Black/African American (n = 105), Latino/Hispanic (n = 141), Native American (n = 2), and Other (n = 1).

^cSome insurance included Medicaid, Medicare/Social security, employment health plan, and Community organization. Pay myself includes family member, friends, no one, pay myself.

^dThe percentage in the right column in this question was calculated based on 234 people who answered YES to the previous Ever-tested question.

^eThis pre-SFHR knowledge of HIV/AIDS status variable is created based on the information from ever told, ever tested, number of HIV test ever taken, and pick-up the last HIV test results.

measure. Similarly, the distribution of coverage for this attribute had more variation than that observed for ILOC/ELOC. Some participants were excluded from the analysis due to the missing information in outcome, exposure, and/or covariate information.

The community detection results could be sensitive to the removal of nodes from the network; therefore, a sensitivity analysis was performed by first conducting community detection, then removing participants with missing information and eliminating isolated participants. As a result of the sensitivity analysis, the number of communities and participants in the SFHR network changed (94 communities with 425 participants). The

Table 3. Unadjusted and adjusted estimated risk differences (RDs) with corresponding 95% confidence intervals (95% CIs) of causal effects of locus of control (external vs. internal) and blame (others vs. self) on likelihood of receiving SFHR HIV test results among 402. Coverage is defined as the probability of internal locus of control in a community. Baseline covariates are included in a community-level propensity score.

Effect	Coverage	Unadjusted		Adjusted		
	($\alpha\%$, $\alpha\prime\%$)	RD	95% CI	RD	95% CI	
Locus of control (external vs. internal)						
Direct	(50, 50)	-0.15	(-0.23, -0.06)	-0.13	(-0.23, -0.03)	
Direct	(70, 70)	-0.14	(-0.25, -0.04)	-0.14	(-0.25, -0.02)	
Direct	(99, 99)	-0.10	(-0.26, 0.06)	-0.10	(-0.26, 0.05)	
Indirect	(50, 70)	-0.04	(-0.07, -0.01)	-0.03	(-0.06, -0.01)	
Indirect	(50, 99)	-0.07	(-0.16, 0.02)	-0.07	(-0.15, 0.00)	
Indirect	(70, 99)	-0.03	(-0.10, 0.04)	-0.04	(-0.09, 0.01)	
Total	(50, 70)	-0.18	(-0.27, -0.10)	-0.17	(-0.27, -0.08)	
Total	(50, 99)	-0.17	(-0.28, -0.07)	-0.17	(-0.28, -0.07)	
Total	(70, 99)	-0.13	(-0.25, -0.01)	-0.14	(-0.26, -0.02)	
Overall	(50, 70)	-0.07	(-0.10, -0.04)	-0.06	(-0.09, -0.04)	
Overall	(50, 99)	-0.10	(-0.18, -0.01)	-0.11	(-0.18, -0.03)	
Overall	(70, 99)	-0.03	(-0.10, 0.04)	-0.04	(-0.10, 0.01)	
Blame (others vs. self)						
Direct	(50, 50)	-0.04	(-0.16, 0.08)	-0.06	(-0.17, 0.04)	
Direct	(70, 70)	-0.04	(-0.16, 0.09)	-0.04	(-0.15, 0.07)	
Direct	(99, 99)	-0.08	(-0.25, 0.10)	-0.07	(-0.23, 0.10)	
Indirect	(50, 70)	-0.00	(-0.05, 0.04)	-0.01	(-0.05, 0.03)	
Indirect	(50, 99)	0.03	(-0.08, 0.14)	0.02	(-0.07, 0.12)	
Indirect	(70, 99)	0.04	(-0.04, 0.11)	0.03	(-0.03, 0.10)	
Total	(50, 70)	-0.04	(-0.16, 0.09)	-0.05	(-0.16, 0.06)	
Total	(50, 99)	-0.04	(-0.20, 0.12)	-0.04	(-0.19, 0.10)	
Total	(70, 99)	-0.04	(-0.20, 0.12)	-0.04	(-0.18, 0.11)	
Overall	(50, 70)	-0.01	(-0.05, 0.04)	-0.00	(-0.04, 0.03)	
Overall	(50, 99)	-0.02	(-0.14, 0.09)	-0.01	(-0.12, 0.09)	
Overall	(70, 99)	-0.02	(-0.10, 0.06)	-0.01	(-0.08, 0.07)	

results for the estimated causal effects of interest were comparable to the main analysis (results not shown).

4 Discussion

We assessed the impact of attitudes among PWID toward HIV/AIDS risk on their own and their neighbors' health-seeking behaviors. PWID with the internal locus of control were more likely receive their HIV testing result, regardless of the internal locus status of other individuals in their community. Individuals who feel they have more control over their health outcomes may feel it is worthwhile to engage in behaviors to improve their health. PWID with the external locus of control in a community with a high coverage of internal were more likely to receive their HIV testing results compared to a low coverage community. Individuals who have friends, partners, or communities with certain health behaviors may engage in these health behaviors not due not only to their own internal motivation but also via reinforcement from their contacts. An intervention that could modify the HIV/AIDS locus of control has the potential to encourage PWID themselves as well as their contacts to preferable health-seeking behavior. Moreover, an intervention aimed at raising coverage from 50% to 70% could yield a comparable impact to an intervention aimed at increasing coverage from 50% to 99%. Such a finding might offer a considerable advantage in terms of resource allocation and effort required to achieve a similar magnitude of outcome. Though attitudes were not a public health intervention in the SFHR study, in terms of social psychology, attitudes could be modified and eventually affect people's behavior [28–31]. Furthermore, even small changes in health-seeking behavior, could have a larger impact over long periods of time, particularly with infectious diseases in networks [6, 16].

There are several limitations to our study. The validity and reliability of the exposure variables as measures of individual attitudes toward HIV/AIDS risk could be further studied. The internal consistency for blame questions suggests less reliability and the coverage of blame attribute had more variability. Future research could address the refinement of both the blame attribute measurement and their coverage thresholds. The development of carefully constructed standardized questionnaires to assess PWID's attitudes toward HIV/AIDS risk could improve this study. There may be missing individuals and connections rendering the observed network likely different from the full underlying network, and future work could be developed to improve ascertainment of edges [35] and address missingness and sampling bias in network-based studies [36]. Another limitation is that this study dataset is dated and the health-seeking behavior of PWID could have changed since the SFHR study was conducted in the early 1990s. However, this work provides insights into attitudes during an emerging HIV epidemic.

There are several future directions for research. First, in this study, the disseminated effect of attitudes was defined as one directional. That is, one individual's exposure affects other's outcome. However, there could be different mechanisms that explain the disseminated effect and future studies could be conducted to better assess this with multiple follow-up visits. A more realistic treatment allocation strategy that allow for correlation of exposure assignment in observational studies could be employed instead of the Bernoulli individual group assignment strategy assumption [32, 33]. Lastly, the

variation in community size could also be considered by using improved estimators such as cluster- and individual-weighted estimators [34]. By understanding attitudes among PWID toward HIV/AIDS on their health-seeking behavior, future interventions could be more effective and sustainable to prevent HIV transmission and improve the HIV continuum of care among PWID.

Acknowledgements. The SFHR data collected by Samuel Friedman with the support from the National Institute on Drug Abuse (NIDA), under National Institutes of Health (NIH) grant 1R01DA006723. The project was supported by the Avenir Award Program for Research on Substance Abuse and HIV/AIDS (DP2) from NIDA award DP2DA046856; and by awards U54GM115677 and P20GM125507 from the National Institute of General Medical Sciences of the NIH. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

1. Centers for Disease Control and Prevention. Hiv and People Who Inject Drugs 2019 [updated August 21, 2019]. <https://www.cdc.gov/hiv/group/hiv-idu.html>
2. Mathers, B.M., Degenhardt, L., Phillips, B., Wiessing, L., Hickman, M., Strathdee, S.A., et al.: Global epidemiology of injecting drug use and HIV among people who inject drugs: a systematic review. *The Lancet*. **372**(9651), 1733–1745 (2008)
3. Ghosh, D., Krishnan, A., Gibson, B., Brown, S.-E., Latkin, C.A., Altice, F.L.: Social network strategies to address HIV prevention and treatment continuum of care among at-risk and HIV-infected substance users: a systematic scoping review. *AIDS Behav.* **21**(4), 1183–1207 (2017)
4. Latkin, C.A., Knowlton, A.R.: Social network assessments and interventions for health behavior change: a critical review. *Behav. Med.* **41**(3), 90–97 (2015)
5. Aiello, A.E.: Invited commentary: evolution of social networks, health, and the role of epidemiology. *Am. J. Epidemiol.* **185**(11), 1089–1092 (2017)
6. Curtis, R., Friedman, S.R., Neaigus, A., Jose, B., Goldstein, M., Ildefonso, G.: Street-level drug markets: network structure and HIV risk. *Soc. Netw.* **17**(3–4), 229–249 (1995)
7. De, P., Cox, J., Boivin, J.F., Platt, R.W., Jolly, A.M.: The importance of social networks in their association to drug equipment sharing among injection drug users: a review. *Addiction* **102**(11), 1730–1739 (2007)
8. Halloran, M.E., Struchiner, C.J.: Study designs for dependent happenings. *Epidemiology*, 331–338 (1991)
9. Hudgens, M.G., Halloran, M.E.: Toward causal inference with interference. *J. Am. Stat. Assoc.* **103**(482), 832–842 (2008)
10. Tchetgen, E.J.T., VanderWeele, T.J.: On causal inference in the presence of interference. *Stat. Methods Med. Res.* **21**(1), 55–75 (2012)
11. Liu, L., Hudgens, M.G., Becker-Dreps, S.: On inverse probability-weighted estimators in the presence of interference. *Biometrika* **103**(4), 829–842 (2016)
12. Benjamin-Chung, J., Arnold, B.F., Berger, D., Luby, S.P., Miguel, E., Colford, J.M., Jr., et al.: Spillover effects in epidemiology: parameters, study designs and methodological considerations. *Int. J. Epidemiol.* **47**(1), 332–347 (2018)
13. Buchanan, A.L., Vermund, S.H., Friedman, S.R., Spiegelman, D.: Assessing individual and disseminated effects in network-randomized studies. *Am. J. Epidemiol.* **187**(11), 2449–2459 (2018)

14. Bandura, A.: Perceived self-efficacy in the exercise of control over aids infection. *Eval. Program Plann.* **13**(1), 9–17 (1990)
15. Allard, R.: Beliefs about aids as determinants of preventive practices and of support for coercive measures. *Am. J. Public Health* **79**(4), 448–452 (1989)
16. Friedman, S.R., Curtis, R., Neaigus, A., Jose, B., Des Jarlais, D.C.: *Social Networks, Drug Injectors' Lives, and Hiv/Aids*: New York: Springer Science & Business Media (2006)
17. Newman, M.: *Networks: An Introduction*: Oxford: Oxford University Press (2010)
18. Kolaczyk, E.D., Csárdi, G.: *Statistical Analysis of Network Data with R*: Springer, New York (2014)
19. Rotter, J.B.: Generalized expectancies for internal versus external control of reinforcement. *Psychol. Monogr. Gen. Appl.* **80**(1), 1 (1966)
20. Blanchard-Fields, F., Hertzog, C., Horhota, M.: Violate my beliefs? then you're to blame! belief content as an explanation for causal attribution biases. *Psychol. Aging* **27**(2), 324 (2012)
21. Grimes, P.W., Millea, M.J., Woodruff, T.W.: Grades—who's to blame? student evaluation of teaching and locus of control. *J. Econ. Educ.* **35**(2), 129–147 (2004)
22. Rozin, P., Royzman, E.B.: Negativity bias, negativity dominance, and contagion. *Pers. Soc. Psychol. Rev.* **5**(4), 296–320 (2001)
23. Saul, B.C., Hudgens, M.G.: A recipe for inference: start with causal inference. add interference. mix well with R. *J. Stat. Softw.* **82**(1), 1–21 (2017)
24. Sobel, M.E.: What Do randomized studies of housing mobility demonstrate? causal inference in the face of interference. *J. Am. Stat. Assoc.* **101**(476), 1398–1407 (2006)
25. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: homophily in social networks. *Ann. Rev. Sociol.* **27**(1), 415–444 (2001)
26. Little, R.J., Rubin, D.B.: *Statistical Analysis with Missing Data*: Hoboken: John Wiley & Sons (2019)
27. Perez-Heydrich, C., Hudgens, M.G., Halloran, M.E., Clemens, J.D., Ali, M., Emch, M.E.: Assessing effects of cholera vaccination in the presence of interference. *Biometrics* **70**(3), 731–741 (2014)
28. Festinger, L.: Cognitive dissonance. *Sci. Am.* **207**(4), 93–106 (1962)
29. Batson, C.D., Polycarpou, M.P., Harmon-Jones, E., Imhoff, H.J., Mitchener, E.C., Bednar, L.L., et al.: Empathy and attitudes: can feeling for a member of a stigmatized group improve feelings toward the group? *J. Pers. Soc. Psychol.* **72**(1), 105 (1997)
30. Batson, C.D., Chang, J., Orr, R., Rowland, J.: Empathy, attitudes, and action: can feeling for a member of a stigmatized group motivate one to help the group? *Pers. Soc. Psychol. Bull.* **28**(12), 1656–1666 (2002)
31. Latkin, C.A., Sherman, S., Knowlton, A.: HIV prevention among drug users: outcome of a network-oriented peer outreach intervention. *Health Psychol.* **22**(4), 332 (2003)
32. Friedman, S.R., et al.: Network research experiences in New York and Eastern Europe: lessons for the southern US in understanding HIV transmission dynamics. *Curr. HIV/AIDS Rep.* **15**, 283–292 (2018)
33. Kim, M., Leskovec, J.: The network completion problem: inferring missing nodes and edges in networks. In: *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, pp. 47–58 (2011)
34. Barkley, B.G., Hudgens, M.G., Clemens, J.D., Ali, M., Emch, M.E.: Causal inference from observational studies with clustered interference. *Ann. Appl. Stat.* **14**(3), 1432–1448 (2020)
35. Papadogeorgou, G., Mealli, F., Zigler, C.M.: Causal inference with interfering units for cluster and population level treatment allocation programs. *Biometrics* **75**(3), 778–787 (2019)
36. Basse, G., Feller, A.: Analyzing two-stage experiments in the presence of interference. *J. Am. Stat. Assoc.* **113**(521), 41–55 (2017)



On the Relation Between Replicator Evolutionary Dynamics and Diffusive Models on General Networks

Rio Aurachman^{1,2}(✉) and Giuliano Punzo¹

¹ University of Sheffield, Sheffield, UK
raurachman1@sheffield.ac.uk

² Telkom University, Bandung, Indonesia

Abstract. Understanding how cooperation spreads across social groups is fundamental in predicting how people will interact with each other in relation to the use and exploitation of the resources they are provided with. When social interactions can be mapped to a network, questions arise about the impact of the connection structure which can benefit from the literature developed for a dynamical systems. One model that is widely used as a model to understand the dynamics of cooperation is the replicator equation. While research has been proposed to adapt that equation to a structured graph, we offer a straightforward approach by benefiting from the networked SI diffusion model and replicator equation to create a replicator equation on a network with state-dependent diffusive constant. This approach can be applied to any network structure and features separation of the game and the information diffusion mechanism. The equilibria towards which the system evolves are here characterised and discussed.

Keywords: Networks · Cooperation · Replicator · Diffusion Model

1 Introduction

Promoting cooperation in populations of selfish individuals is an extensive field of research. Numerous studies have used well-mixed or fully connected networks as a model [15]. At the beginning of the 21st century, it was recognised that some systems are well suited to be modelled as networks, which prompted the investigation of the evolution of cooperation in structured networks [8].

A completely connected network, where individuals represented by the vertices of the graph and the edges specify who interacts with whom, can also represent a well-mixed population in which everyone can interact with everyone else [5]. However, this does not exploit the flexibility of using a network to capture the variety of social connections [13] influencing the game. Because of that, other kinds of simple networks were proposed in this context, including trees, star networks [16], bipartite and line graphs [3].

A lattice network was proposed in [18, 22] as the model, which was appropriate to capture emerging behaviour in the classical prisoner dilemma [10], as also

mentioned in [4]. The more frequent use of lattice networks followed from using a uniform, yet not fully mixed, environment in which most of the models originating in statistical physics can be used to describe the emergence and evolution of cooperation [12]. Despite their dissimilarity to real social networks, they offer a beneficial starting point to investigate how structure affects the development of cooperation [12]. Beyond regular graphs [4], more complex network structures have been considered, including the random geometric network [5] and the scale-free network [17]. Although more rare, some games dynamics and control studies have considered generic network structures [16].

Replicator dynamics is widely used in the evolution of the strategy and the literature offers a number of enhancement and variation to the original formulation. It works on the assumption that the strategies adopted by players who are successful in the game are adopted by players who are less successful. The probability of switching strategy is a function of the difference in player's payoff average between each strategy in the game. The strategy of players with greater fitness or reward is replicated by those with lower achievement in the game as the game is played repeatedly [20].

Related to structure, the dynamics of the conventional replicator assumes that the players are engaged in a game and the adoption of global strategies. Each player engages with a representative sample of the population, and if a strategy produces a payoff that is higher than the average, the strategy will be chosen and spread [6]. Understanding that the game could be played on a structured population with a general network, the individual best strategy may be different from player to player, as each engage in games with a different set of neighbours. This, motivates us to consider a multiplier, function of the perceived payoff from choosing a strategy, to model its dynamics in a general network.

Recently, there has been an increasing interest in the research community in studying replicator equation on graph. One of the graph structures is the regular graph [11] which has been considered also in multiplex graphs [14]. Another approach is to modify the regular graph into a modular regular graph [2]. Beyond that, the replicator equation has been used in conjunction with different types of network, including random regular networks, Erdos-Rényi networks, Watts-Strogatz small world networks, and Barabási-Albert scale-free networks [19].

Application to even more general graphs has been attempted in [7][21]. This offers some benefit to understanding the impact of network structure on evolutionary dynamics. However, It is quite popular to assume the strategy change happens simultaneously, with consequent overlooking of dynamic aspects of the game and information diffusion. Understanding this potential for contribution, we develop a model to capture evolutionary dynamics that can be implemented in any population structure with some additional features.

The model mimic the shape of the usual susceptible-infected (SI) diffusion model on a network. Relating it to the structure of the replicator equation, the SI model's infection rate is defined based on the payoff matrix of the replicator model and the payoffs the players acquire from it. This combination of the repli-

cator equation and the susceptible-infected (SI) model hints at mechanisms of information diffusion in the evolution of the strategy on a general network.

There are some novel benefits of the combination. First, it becomes possible to separate the game and information diffusion process using this new model. The effect of the game and the diffusion process toward cooperation can be better understood separately. As this can also be extended into multilayer network games, it opens up the development of a novel model that uses other diffusion processes or opinion dynamics and other game types.

2 The Model

The network is modelled as a graph $G = (V, E)$. The node v_i is in the set of nodes V . It can be written as

$$v_i \in V. \quad (1)$$

E is a set of ordered pair of nodes, also known as edges, that is $e \subseteq v \times v$.

Every node will have specific interactions with other nodes based on the connection structure, as captured by the adjacency matrix $A = [a_{ij}]$. It is necessary to add the new variable of the neighbour set H_i , which indicates the neighbour of i .

We consider a Public Goods game with continuous strategy set $x := [0, 1]$ mapping the extent by which each node is contributing (hence cooperating) to the public good. With x_i indicating the amount by which a node cooperates, $1 - x_i$ will be the proportion of defection. This approach is often referred to as individual-based mean field [9], where the state variable can be intended as a continuous variable or as the probability that the node is in one of the binary states 0 (defection) or 1 (cooperation). In each round of the game, interactions will incur some payoff for each player or agent, which is the function of $x_i(t)$ and $x_j(t)$ as written in Equation

$$\pi_i(t) = f(x_i(t), x_j(t)) \text{ where } v_j \in H_i. \quad (2)$$

After receiving the payoff, each player will update their strategy. This change is modelled in the concept of strategy update dynamics, which depend on strategy and payoff in the neighbourhood. It is written as follows;

$$\dot{x}(t) = l(\{x_j(t), x_i(t), \pi_i(t), \pi_j(t) : i, j \in H_i\}), \quad (3)$$

where the initial condition is $x_i(0)$ and the control is $u_i(t)$ for each agent $i \in V$.

3 Replicator Equation Model Development

3.1 Related Model

Before the game dynamics is introduced, let us introduce the dynamics of a well-known epidemic model, known as Susceptible-Infected (SI) model. In fact, we

shall note how the diffusion of cooperation within the network can be modelled as a pandemic, and in particular by means of the SI model. For a constant size of the population, only the fraction of infected can be considered, with the susceptible being the complement to 1 to it. The differential model in scalar and vector form is [1]

$$\dot{x}_i(t) = \beta(1 - x_i(t)) \sum_{j=1}^n a_{ij} x_j(t), \quad (4)$$

$$\dot{\mathbf{x}}(t) = \beta(I_n - \text{diag}(\mathbf{x}(t))) \cdot \mathbf{A}\mathbf{x}(t). \quad (5)$$

The a_{ij}^k has a value 0 or 1, which 1 means that there is an edge between v_i and v_j in the layer k . This is the networked version of the basic SI model in a well-mixed population, which is expressed as

$$\dot{x} = \beta \cdot (1 - x) \cdot x. \quad (6)$$

It comes from the idea that the cooperation condition is transformed from its fraction of defection multiplied by its interaction with the neighbour's fraction of cooperation, summed up.

Understanding the nature of replicator equation and network SI model, there is an opportunity to combine these two functions to model replicator equation in a general network. The connection between two models is as follows:

- Differential equations can represent the replicator and SI epidemic models with continuous time dynamics,
- The population is divided into many segments or groups in both models, often called compartments in the SI model,
- In both models, the rates of change depend on how people in the population interact with each other,
- Both models have equilibrium points that represent stable or steady-state conditions.
- Both models can be examined using related mathematical methods, such as stability analysis, to determine how the system behaves around equilibrium points.

Considering the similarity and connection between two models, the following section will explain the combination process of the two models.

3.2 Combining the SI Model and Replicator Equation

The payoff matrix that will be used for cooperation and defection is

$$\begin{bmatrix} P & Q \\ R & S \end{bmatrix}. \quad (7)$$

The basic equation of replicator is [5]

$$\dot{x} = x \cdot (f_c - \bar{f}) \quad (8)$$

where,

$$\bar{f} = x \cdot f_c + (1 - x) \cdot f_d . \tag{9}$$

The equation introduces new variables f_c , f_d , and \bar{f} which mean the average payoff of the cooperator, the defector, and the population, respectively. Based on Eqs. 9 and 8, the modified replicator equation will be:

$$\dot{x} = x \cdot (f_c - x \cdot f_c - (1 - x) \cdot f_d), \tag{10}$$

or,

$$\dot{x} = x \cdot (1 - x) \cdot (f_c - f_d). \tag{11}$$

The equivalence between an SI model in Eq. 6 and the replicator Eq. 11 becomes evident when considering

$$\beta := f_c - f_d. \tag{12}$$

Understanding the similarity between the networked SI model in Eq. 6 and the replicator equation for the evolutionary game in Eq. 11, the networked evolutionary game can be modelled as a cooperation contagion process. Rather than representing infectious, the variables x represent cooperation and $1 - x$ represent defection. The β value is probability of contagion of cooperation, which represent how superior cooperation compared to defection, and influenced by the amount of cooperator (namely, x)

3.3 The Network Structure of Information Diffusion

Using the concept from [1], $(1 - x) \cdot x$ can be a diffusion process in the SI model. Therefore, it means that $1 - x$ is a percentage of defection in the nodes. It may change by the proportion of cooperation x with its neighbour, multiplied by the contagion factor β . The new equation will be

$$\dot{\mathbf{x}}(t) = \beta(I_n - \text{diag}(\bar{\mathbf{x}}(t)))A\bar{\mathbf{x}}(t) = (f_c - f_d)(I_n - \text{diag}(\bar{\mathbf{x}}(t)))A\bar{\mathbf{x}}(t). \tag{13}$$

where $\bar{\mathbf{x}}(t)$ is the average value of \mathbf{x} according to its neighbour at time t or

$$\bar{\mathbf{x}}(t) = \text{diag}^{-1}(A\mathbf{1})A\mathbf{x}(t). \tag{14}$$

Because the information dissemination process will include not only the neighbour but also self-reinforcing belief loops A will be given by the adjacency matrix added with the identity matrix or $A = A^* + I$.

3.4 Network Structure on Game

As has been introduced in Eq. 12, we assume that the rate at which cooperation spreads depends on the expected payoff difference between cooperation and defection. In particular, in well-mixed population, a player will expect a payoff from cooperating equal to $f_c = x \cdot P + (1 - x) \cdot Q$, which x considers the fraction of population expected to cooperate and $1 - x$ expected to defect. Likewise, the expected reward from defection will be $f_d = x \cdot R + (1 - x) \cdot S$.

Introducing the role of network structure on β , will modify the equation to become:

$$\beta_i = f_c - f_d = \sum_j [x_j \cdot P + (1 - x_j) \cdot Q] - [x_j \cdot R + (1 - x_j) \cdot S]. \quad (15)$$

This can be simplified to become

$$\beta_i = \sum_j [x_j \cdot (P - R) + (1 - x_j) \cdot (Q - S)]. \quad (16)$$

Equation 16 shows the possible role of the network structure in the game. The variable β represents the cooperator payoff minus the defector payoff. Equation $x_j(P - R)$ means that the player i has the value β which is the result of payoff of cooperation (P) minus payoff of defection (R), multiplied by x_j . The variable x_j is the fraction of cooperation of the neighbour of i . The same applies for $(1 - x_j)(Q - S)$ but for the defecting neighbour. When the neighbour is related to the network structure, Eq. 16 can be modified into

$$\beta_i = \sum_j a_{ij} [x_j \cdot (P - R) + (1 - x_j) \cdot (Q - S)]. \quad (17)$$

In the matrix form it will become,

$$\beta = [A(\mathbf{x} \cdot (P - R) + A(\mathbf{1} - \mathbf{x}) \cdot (Q - S))]. \quad (18)$$

Then the complete equation from 18 and 13 will be

$$\dot{\mathbf{x}}(t) = \bar{\mathbf{x}} - \mathbf{x} + [A\mathbf{x} \cdot (P - R) + A(\mathbf{1} - \mathbf{x}) \cdot (Q - S)] \odot (I_n - \text{diag}(\bar{\mathbf{x}})) \cdot A\bar{\mathbf{x}}(t), \quad (19)$$

where \odot is the hadamard product between two vectors or two matrices.

4 Results

We can then offer some results about the system's equilibrium through numerical simulations. After experimenting with other kinds of networks which produce consistent results, We present a result from the Erdos-Renyi network with ten nodes. The initial value of \mathbf{x} are equally spread between 0 and 0.9 A key difference with the classical SI dynamics is in the value of $\beta = f_c - f_d$, which in this case is between -1 and 1. This is because in the SI model, β is a probability of transformation $1 - x$ into x . Therefore, it should not be greater than 1 for positive value of \dot{x} (Increasing x) and should not be less than 1 for negative value of \dot{x} (decreasing x). In other words $-1 \leq \beta \leq 1$. Therefore, $|Q - S| \leq 1$ and $|P - R| \leq 1$.

The simulation is set as an anticonoordination game. For the anticonoordination game, Q and R in the payoff matrix should be the highest value. Therefore, $0 \leq \max\{Q, R\} - \min\{P, S\} \leq 2$

Looking at Eq. 19, the payoff matrix component that affects \dot{x} is not $P, Q, R,$ or S individually, but $P - R$ and $Q - S$. It is consistent that if the value of P and R is changed but the value of $P - R$ remains the same, the equilibrium point will not change. The same applies for $Q - S$. The equilibrium point will remain the same even if the pay-off matrix is changed, as long as $P - R$ and $Q - S$ are the same.

Thus, we suggest that the equilibrium point is related to the ratio between the difference of P, R and the difference of Q, S . Or in other words,

$$\mathbf{x}(T) \propto \left| \frac{Q - S}{P - R} \right|. \tag{20}$$

The simulation also shows that the higher $\left| \frac{Q - S}{P - R} \right|$ will make the equilibrium of \mathbf{x} higher. The result of the simulation using some value of $\left| \frac{Q - S}{P - R} \right|$ is shown in Fig. 1.

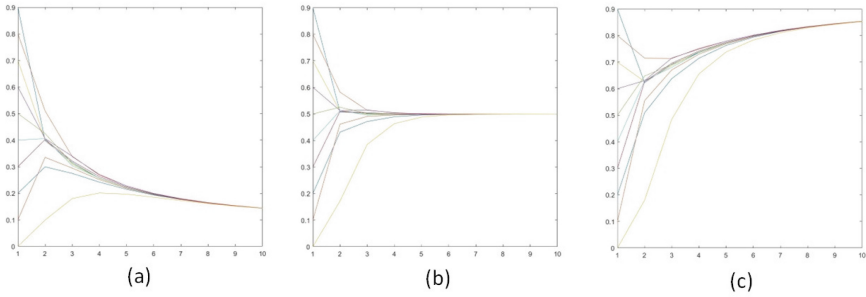


Fig. 1. Value of \mathbf{x} using dynamical equation Eq. 19 where value of $\left| \frac{Q-S}{P-R} \right|$ equal to (a) 0.1, (b) 1, and (c) 10.

5 Conclusion

The replicator dynamics, originally implemented in a well-mixed population, is one of the most widely used methods as is or developed with some enhancement and variation. We propose a new approach to model the replicator equation in a general graph. Although some of the previous findings merge the game and strategy diffusion process, we try to separate those. Taking advantage of the replicator equation and SI model, the mechanism of the game and strategy diffusion process can be better understood.

This modelling approach opens up possibilities for future development. With the separate mechanism of game and information diffusion, it can be extended into a two-layer network. The diffusion process can also be exchanged with another spreading mechanism in the network, such as the opinion dynamic and another more complex model and mechanism.

Acknowledgement. This work funding was supported by Puslapdik BPI (Pusat Layanan Pendidikan Beasiswa Pendidikan Indonesia) Ministry of Education, Culture, Research and Technology of Indonesia, LPDP (Lembaga Pengelola Dana Pendidikan) Ministry of Finance of Indonesia, and Telkom University.

References

1. Bullo, F.: Lectures on Network Systems (2018)
2. Cassese, D.: Replicator equation on networks with degree regular communities. *Appl. Network Sci.* **3**, 29 (2018). <https://doi.org/10.1007/s41109-018-0083-2>
3. Das, S., Eksin, C.: Approximate submodularity of maximizing anticoordination in network games. In: 2022 IEEE 61st Conference on Decision and Control (CDC), pp. 3151–3157 (2022). <https://doi.org/10.1109/CDC51059.2022.9993180>
4. Du, J., Wu, Z.: Evolutionary dynamics of cooperation in dynamic networked systems with active striving mechanism. *Appl. Math. Comput.* **430**, 127295 (2022). <https://doi.org/10.1016/J.AMC.2022.127295>
5. Ghivarello, S., Antonioni, A.: Coevolution of individual perception and cooperative behavior in the norm compliance dilemma (2022)
6. Hilbe, C.: Local replicator dynamics: a simple link between deterministic and stochastic models of evolutionary game theory. *Bull. Math. Biol.* **73**, 2068–2087 (2011). <https://doi.org/10.1007/s11538-010-9608-2>
7. Li, J.Y., Wu, W.H., Li, Z.Z., Wang, W.X., Zhang, B.: Data-driven evolutionary game models for the spread of fairness and cooperation in heterogeneous networks. *Front. Psychiatry* **14**, 1131769 (2023). <https://doi.org/10.3389/fpsy.2023.1131769>
8. Llebberman, E., Hauert, C., Howak, M.A.: Evolutionary dynamics on graphs. *Nature* **433**, 312–316 (2005). <https://doi.org/10.1038/nature03204>
9. Miegheem, P.V., Omic, J., Kooij, R.: Virus spread in networks. *IEEE/ACM Trans. Networking* **17**, 1–14 (2009). <https://doi.org/10.1109/TNET.2008.925623>
10. Nowak, M.A., May, R.M.: Evolutionary games and spatial chaos. *Nature* **359**, 826–829 (1992). <https://doi.org/10.1038/359826a0>
11. Ohtsuki, H., Nowak, M.A.: The replicator equation on graphs. *J. Theor. Biol.* **243**, 86–97 (2006). <https://doi.org/10.1016/j.jtbi.2006.06.004>
12. Perc, M., Gómez-Gardeñes, J., Szolnoki, A., Floría, L.M., Moreno, Y.: Evolutionary dynamics of group interactions on structured populations: a review. *J. R. Soc. Interface* **10**, 20120997 (2013). <https://doi.org/10.1098/rsif.2012.0997>
13. Perc, M., Jordan, J.J., Rand, D.G., Wang, Z., Boccaletti, S., Szolnoki, A.: Statistical physics of human cooperation. *Phys. Rep.* **687**, 1–51 (2017). <https://doi.org/10.1016/j.physrep.2017.05.004>
14. Requejo, R.J., Díaz-Guilera, A.: Replicator dynamics with diffusion on multiplex networks. *Phys. Rev. E* **94**, 022301 (2016). <https://doi.org/10.1103/PhysRevE.94.022301>
15. Riehl, J.R., Cao, M.: Towards control of evolutionary games on networks. In: 53rd IEEE Conference on Decision and Control, pp. 2877–2882 (2014). <https://doi.org/10.1109/CDC.2014.7039831>
16. Riehl, J.R., Cao, M.: Towards optimal control of evolutionary games on networks. *IEEE Trans. Autom. Control* **62**, 458–462 (2017). <https://doi.org/10.1109/TAC.2016.2558290>
17. Santos, F.C., Pacheco, J.M.: Scale-free networks provide a unifying framework for the emergence of cooperation. *Phys. Rev. Lett.* **95**, 098104 (2005). <https://doi.org/10.1103/PhysRevLett.95.098104>

18. Szolnoki, A., Perc, M.: Reward and cooperation in the spatial public goods game. *Europhys. Lett.*= *Europhys. Lett.* **92**, 38003 (2010). <https://doi.org/10.1209/0295-5075/92/38003>
19. Tan, S., Wang, Y.: Graphical nash equilibria and replicator dynamics on complex networks. *IEEE Trans. Neural Networks Learn. Syst.* **31**, 1831–1842 (2020). <https://doi.org/10.1109/TNNLS.2019.2927233>
20. Taylor, P.D., Jonker, L.B.: Evolutionary stable strategies and game dynamics. *Math. Biosci.* **40**, 145–156 (1978). [https://doi.org/10.1016/0025-5564\(78\)90077-9](https://doi.org/10.1016/0025-5564(78)90077-9)
21. Veelen, M.V.: The replicator dynamics with n players and population structure. *J. Theor. Biol.* **276**, 78–85 (2011). <https://doi.org/10.1016/j.jtbi.2011.01.044>
22. Wang, S., Chen, X., Xiao, Z., Szolnoki, A.: Decentralized incentives for general well-being in networked public goods game. *Appl. Math. Comput.* **431**, 127308 (2022)

Dynamics on/of Networks



SMART CONTRACTS Based Peer to Peer Communication in Blockchain: A Decentralized Approach

Satya Bhushan Verma^{1(✉)}, Abhay Kumar Yadav¹, Bineet Kumar Gupta¹, and Sanjay Gupta²

¹ Shri Ramswaroop Memorial University, Barabanki 225001, India
satyabverma1@gmail.com, abhaybbdnitm01@gmail.com,
bkguptacs@gmail.com

² Corporate Vice President, HCL Technologies Limited, Chennai, India
san-jay_gupta@hcl.com

Abstract. The Electronic messaging is a widely used network application, and the user's authentication is a necessary aspect. Commonly employed approaches include PKI and S/MIME encryption protocols for email, but have many security threats, such as EFAIL and attack MITM attack. The innovative property of blockchain minimizes these threats and provides decentralized sensitive operations along with high security. It terminates the need of trusted intermediaries. Smart contracts are implementation of second generation of blockchain technology. The objective of our work is to provide a secure peer to peer messaging solution. In this paper, we explain how blockchain would ensure secure communications, and also, we have designed a model for messaging system that ensures good performance and data security which verifies the user identities and their public keys, and also validate the user certificate. The proposed messaging system is decentralized in nature and allows secure messages exchange.

Keywords: Decentralized Approach · Blockchain · Peer to Peer · Communication

1 Introduction

Centralized mixing services are mainly dependent on a trusted or semi-trusted third party to mix the transaction values of multiple users and produce the output to the corresponding addresses so it become difficult for the attackers to link the input and output addresses of the transaction. In centralized systems, all requests pass through a central authority, which decides whether to accept or reject the request. This approach eases access control administration because a single entity can manage all configurations. Even though they have higher efficiency, they might suffer from risk of a single point failure like most other centralized systems. Another drawback is that it requires central authority compulsory permission for any execution. Also, it has lower transparency in the network. So, an alternative approach to the centralized mixing, i.e., decentralized mixing, are needed to be explored more, as it would benefits all users with terminating

the problem of single point failure of centralized mixing. Also, decentralized mixing doesn't need any mixing fees.

Peer to peer (P2P) systems eliminates many problems of traditional client-server approaches, however these features introduced newer issues such as establishment the trust relationship in P2P networks.

Blockchain consist of a distributed database network responsible for recording all transactions that occurs in a network. Blockchain was originally introduced for Bitcoin (a peer-to-peer digital payment system), later on a wide range of decentralized applications were created. Blockchain technology serves as a potential solution to various problems [13]. It helps in achieving data integrity, tamper-resistance based on cryptography. The blockchain is decentralized, and do not need centralized authority to approve transactions. All participants must reach a consensus for secure transactions validation, and earlier records cannot be altered. It is very costly to alter the previous record [6] (Fig. 1).

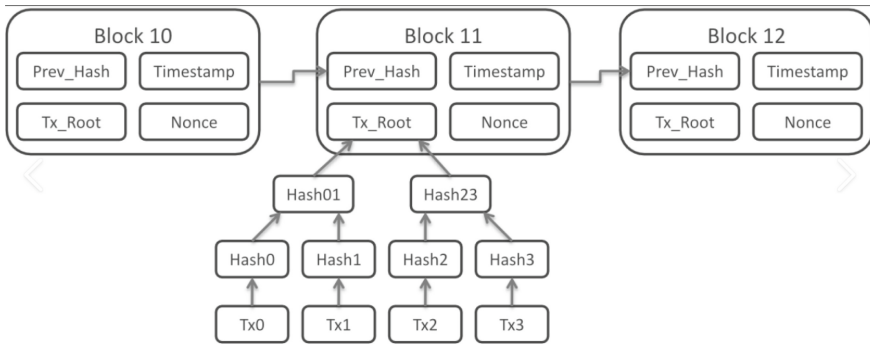


Fig. 1. A simple blockchain illustration

The first generation of blockchain technology were termed as Cryptocurrencies. They are the digital currencies being derived from peer-to-peer network and cryptographic techniques. One of the earliest and popular examples of cryptocurrency was Bitcoin, an electronic payment system that permits two different parties to digitally transfer money amongst users without the need of any intermediate network [3].

The second generation consists of other blockchains networks such as Ethereum that permits complex distributed applications creation more than just the cryptocurrencies. Smart contracts, been discussed in the following section, is the main component of this generation [15]. Ethereum platform is the very popular for smart contracts creation. Ethereum is a public blockchain with a built-in Turing-complete language to allow writing any smart contract and decentralised application.

A smart contract is a computer protocol created with purpose of digitally facilitating and enforcing any negotiation of a contract in any unreliable environment Comparing with traditional contracts, smart contracts are independent of the third-party requirement, so have low transaction costs. Smart contract code means “code that is stored, verified and executed on a blockchain” [10]. The smart contract body has account balance, a

private storage and executable code. Smart contracts run on peer-to-peer network with an algorithmic code.

There are two types of smart contracts, deterministic and non-deterministic smart contracts [16]. A deterministic smart contract is independent of any information from an external party (from outside the blockchain). Non-deterministic smart contracts are dependent on information from any external entity. For example, a contract that works on current information such as weather change, which is not available on the blockchain.

There are different blockchain platforms that can be utilised to develop smart contracts. These include bitcoin - a publicly available blockchain platform for processing cryptocurrency transactions but have very less compute capability [2]. Other include NXT- an open source blockchain platform based on proof-of-stake consensus protocol public blockchain platform that uses smart contracts as templates [1] and Ethereum - a decentralized open source blockchain which support advanced and customized smart contracts [7]. Ethereum's is the most commonly used blockchain because its language supports Turing-completeness feature which execute exactly as programmed, and can be accessed anytime, anywhere in the world.

2 Literature Review

Existing literature consists of substantial amount of research based on centralized access control. However, research on decentralized access control in a collaborative environment is scarce.

An overview of various scripting languages been used in blockchain networks was proposed by Seijas et al. [14] in 2017 that introduced Bitcoin, Ethereum and Nxt application in implementation of smart contract. They also emphasize some potential security problems of smart contracts such as re-entrancy, runtime exceptions, incomplete preconditions handling, unpredictable state, immutable bugs, non-randomness and elaborate question on scripting language to be Turing complete. They also proposed technical solutions to verify or enhance scripting languages.

Popular implementation of smart contract in improving consumer protection was given by Fairfield [5]. They proposed various applications for smart contract in the blockchain network such as in online contracting. Till now, the companies can only show the price term online, the consumers can either accept or decline the amount. Using smart contracts there is possibility to bargain about the terms.

Werbach et al. [17] in 2017 explained potential role of startups in smart contract development that might replace contract law present used nowadays. The authors debated the question on whether smart contracts can be used in laterally with the law. In the end, they concluded that new opportunities will arise in commercial fields but will not alter any existing contract law.

One of the first decentralized mixing services was CoinJoin [8] which was proposed by Maxwell in 2013, which enabled users to mix their own coins in a self-organized way and not depending on any third party. At the beginning of mixing approach, a set of negotiation process were conducted amongst the set of paying authority, which provides confirmation to recipient whom they wish to make the payment. After that a transaction that ensures all the input/output pairs of sender and receiver were generated and were

checked by users to ensure their payment destination was properly encapsulated. If the transaction managed to pass the verification process by all the payers, they would further sign the transaction jointly and finally published it via blockchain. Comparing with other centralized methods, CoinJoin minimised the deduction risk of transaction linkage due to outer/inner attackers and eliminates the coin theft problem. However, there were still some problems present in the CoinJoin. At the time of negotiation, the users that are involved in the coin mixing might be able to know the information of other clients.

One of the earliest approaches for secure transactions between users was BitDNS[11], its success inspired many more numerous approaches. Namecoin is the first system to be built on decentralized naming system using blockchain, Satoshi believed that BitDNS should use its independent blockchain and even offered the first proposal for merged mining as a way to secure blockchain. Namecoin was also amongst the first solution to Zooko's triangle producing a naming system that is simultaneously secure, decentralized, and human-meaningful. Namecoin suffers from 51% attacks [4], because of its insufficient computing power. Blockstack uses the existing internet transport layer (TCP or UDP) and underlying communication protocols and focuses on removing points of centralization that exist at the application layer. The underlying blockchain of Bitcoin limits the performance of Blockstack. [18].

Certcoin [9] removes central authorities and uses the blockchain Name coin as a distributed ledger of domains and their associated public keys. Every certcoin user stores the entries blockchain, and this causes two problems, the latency for the controller and the security problems of merged mining used by Namecoin. Emercoin [12]: Blockchain-based PKI that doesn't remove central authorities but uses Blockchains to store hashes of issued and revoked certificates. Emercoin has the side benefit of optimizing network access by performing key and signature verification on local copies of the blockchain. However, all of these systems faced the same barrier. This consensus involves significant energy, delay, and computation overhead because of high resource demand for solving the Proof of work.

3 Methodology

The primary goal of our approach is to secure communications between entities of the network. The proposed model is to use the blockchain to validate the user's identity and to ensure trust between users for exchanging messages with a high level of security. Each user must communicate only with the user's identity validated by the smart contract, and consider every other interaction as malicious. Each user who is interested in communication needs to register their identity and public key in blockchain (Fig. 2).

To achieve such a system, we use Ethereum public blockchain that implements smart contracts. Ethereum provides a decentralized platform that can be used to deploy decentralized value transfer applications or Dapps. These Dapps are famously called smart contracts that run exactly as specified in their code without any third-party interference or censorship. This is possible as these smart contracts run on top of a blockchain [15].

A unique address of 20 bytes is assigned to each contract. A deployed contract into the blockchain cannot be changed. For running a contract, a transaction is sent to contract address of the user. The transaction is then executed by every node (called miners) in the network to reach an output consensus. The contract's state is updated accordingly.

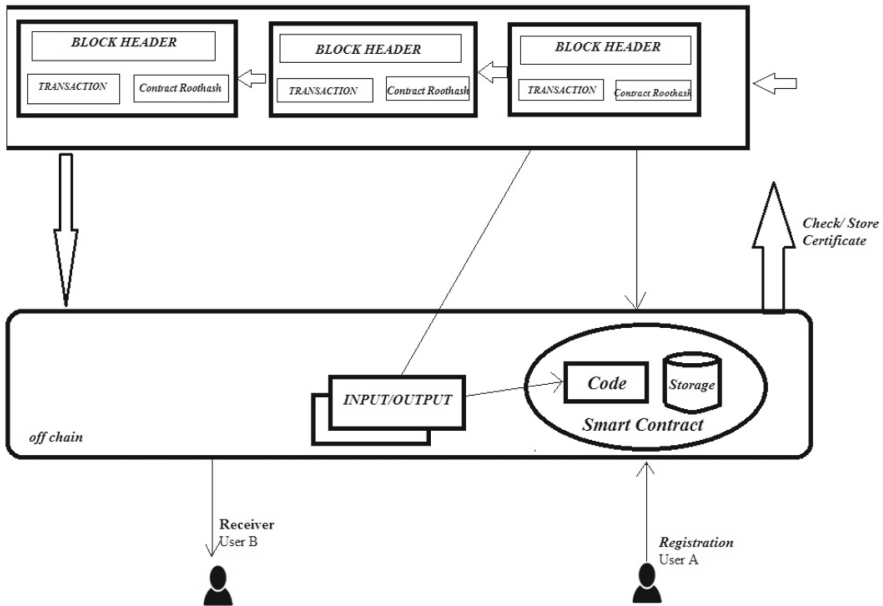


Fig. 2. System Architecture of proposed model

3.1 Data Validation using Smart Contracts

A smart contract is a computer application that would be entered into the smart contract blockchain which assists the data owner throughout verifying the auditing outcome as well as immediately transferring digital assets depending upon that. A smart contract seems to be a series of digital agreements that are immediately carried out by computer networks. As illustrated in Fig. 3, smart contracts implemented on a blockchain comprise a payment & storing technology as well as a total state structure. Smart contracts allow a complicated series of triggered electronic obligations to be successfully fulfilled based on the contract users' wants. The asset condition gets refreshed when a payment update is received, and the smart contract gets activated to analyze the state machine. When several of the trigger requirements are fulfilled, the smart contract conducts the trade as per the predefined data and informs the clients.

P2P networking allows nodes can exchange system resources like computing power or memory size. Such shared resources are given via the connectivity is provided freely through other peer nodes without the need to go through a middleman. As a result, when contrasted to the user approach, the P2P approach would be more decentralized and also has superior flexibility and scalability. Furthermore, information is dispersed between nodes without first passing through a centralized link, considerably reducing the risk of monitoring and the loss of a user's private details.

3.2 Proof-of-Work (PoW)

The PoW operation is performed by looking for a value that seems to have a hash beginning with several zero bits is hashed. It is done by inserting a nonce (doing work) towards the original value till the generated hash contains the specific quantity of zero bits. Once this nonce was determined as well as the proof of work is met, the block cannot be modified without repeating all works for that block and all subsequent blocks.

Excluding the initial block formed by the network (genesis block), every block includes a hash that comprises the preceding block’s hash plus the nonce associated with producing the appropriate zero bits, as shown in Fig. 4. The genesis block is the exception since it has no preceding block to reference: its hash seems to be all zeros. Finally, the bitcoin payments can be verified and preserved without the need for a centralized authority according to PoW.

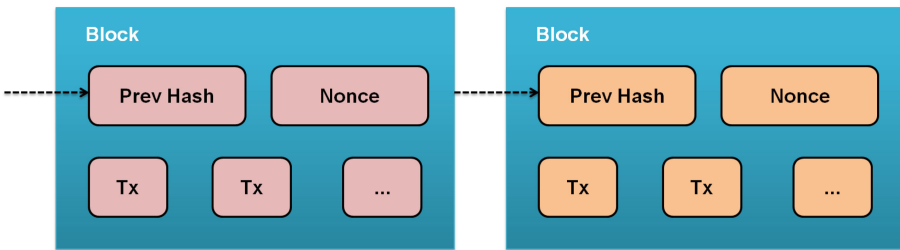


Fig. 3. PoW Function

3.3 Cluster-Authority Selection (CAS)

In the beginning, every node is generally placed and has varying energies. To identify a cluster authority, every node is given a threshold level. Every node receives k_0 as follows from the method.

$$k_0 = \sqrt{\frac{\epsilon f}{\epsilon m}} \tag{1}$$

Here, ϵf = Free breaks, which depend on the range between both the sensor node and the sink, and ϵm = Multiple paths, which depend on the range between both the sensor node and the sink.

The k_0 is employed to estimate G_a .

$$G_a = \frac{1}{0.765} * \sqrt{\frac{2}{M\pi}} * k_0 \tag{2}$$

Here, $(1/0.765)$ = range of the base station, M = amount of nodes, and G_a = estimation of actual threshold level.

Every node determines if it is a cluster authority by calculating the threshold level. The below equation was used to calculate the actual threshold level for every node.

$$T_s = \frac{G_a}{1 - G_a * \left(d \bmod \left(\frac{1}{G_a} \right) \right)} \left[\frac{E_i}{E_{av}} + \left(d \bmod \frac{1}{G_a} \right) \left(1 - \frac{E_i}{E_{av}} \right) \right]$$

Here, d = network's current round, E_i = sensor's energy, and E_{av} = average value of overall network energy.

Every node's threshold value is matched with that of other nodes. It will become the node that has the greatest threshold level that'll be chosen as the cluster authority. Algorithm 1 depicts the procedure of the CAS technique.

Algorithm 1: CAS procedure

```

Function CA Selection ()
i=nearest CA peer () & bigger weight
if
CA ≠ i then
CA=i
Connect to (i)
else if
do nothing()
Forward(CAINV)
end if

```

3.4 Blockchain Registration

At first, the initial sender begins the communication with public and private key and then derives the hash identity. He keeps the private key safe and tends to register its hash identity with the public key into the blockchain. The public key registers its identity after being verified and validate by the network.

1. Each public key in the blockchain have an associated timestamp with it
2. The sender signs transactions with the corresponding private key and transfers to the blockchain
3. The miner checks that the ID of sender is unique and has never previously been registered in the network
4. If verified correctly, the sender's certificate is then stored on the blockchain, with the information such as Id, Public key, the validity of the public key and timestamp
5. Similar registration process is followed by all active nodes in blockchain.

3.5 Smart Contract-Based Verification

When another user B wants to send a message to initial first user, he only presents the public key ID already present on the blockchain along with the ID of first user with a time-stamp T to the blockchain. Each message must include the time.

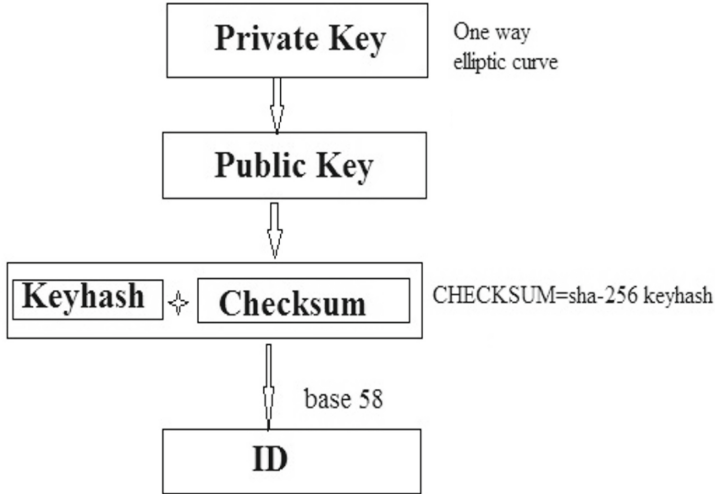


Fig. 4. Identity generation in blockchain smart contract

1. User B sends the transaction; the smart contract accepts the request from the user B in the network.
2. The smart contract checks if the request exists on the blockchain. The Smart contract reads and parses the two users A and B recordings.
 - a. It performs a Lookup for the output return true.
 - b. Once the public keys are verified, the Smart contract verifies the validity of the timestamp and the signature of the transaction.
 - c. Finally, the smart contract validates the request and return true.
3. User B is then permitted to sends a transaction to user A address
4. Next, user A checks the transaction with the private key and then sends it to B address
5. After receipt of the transaction by second user B, verification will be performed again, and mutual authentication will be established between the two entities.

3.6 Send/Receive Message

Once mutual authentication is established, A uses B’s public key and A’s private key to generate a shared secret. The shared secret can be generating using the Elliptic-curve-Die-Hellman (ECDH) to encrypt Message. Using ECDH we can generate a shared key in the network [12] which can be used in message encryption using a symmetric key algorithm

4 Advantages and Limitations

We have used the blockchain as a distributed identity along with their associated public. It has eliminated the need of central authorities (CA). Blockchain has been used to store public keys, digital signature, and information about the peers in the network. The proposed system is trust worthy, transparent and traceable

Once established, the smart contract code works accurately as programmed. This serve as one of the main advantages of the blockchain platform, the code cannot be falsified and will interact as promised, and it would never suffer a downtime. Our proposed system ensures confidentiality, message integrity, authenticity and reliability.

In blockchain, the smart contract is executed sequentially that affect the performance of blockchain negatively. With the increasing number of smart contracts, the blockchain will not be able to scale. In practice, it is impossible to modify an existing contract in the blockchain after it has been registered in it. The design phase of these contracts will, therefore, require special attention to avoid any future disappointment.

5 Conclusion and Future Scope

We have proposed a simple step by step method for establishing secures communication in blockchain network with the help of security properties of blockchain. It demonstrates how we can use the blockchain network to provide a solution to problems of centralized PKI. This decentralized architecture offers fault tolerance, redundancy, and transparency in the network. Further, we intend to implement an upgraded architecture with smart contracts to validate, store and revoke the certificate on a public blockchain. That certificate would store user address and its public key, smart contract issuer address, and also store all data in any off-chain network.

Research on smart contracts has already produced substantial findings, but the discourse is still at an early stage, as indicated by the exponential growth in the number of publications. Blockchain technology and smart contracts are highly innovative environments that are developing rapidly outside of scientific research. This becomes clear not least from the social network analysis. Neither of the two largest nodes are peer reviewed publications. This suggests that researchers should not hesitate to engage in active projects and grey literature outside of academic research to monitor current developments. While the peer review provides the desired security and seal of approval for this analysis, the process also involves a considerable time lag. For this reason, future research should always bear in mind the blockchain/smart contract ecosystem and the grey literature on the topic.

References

1. Lewis, A gentle introduction to smart contracts. <https://bitsonblocks.net/2016/02/01/a-gentle-introduction-to-smart-contracts>
2. Andrea Corbellini. <https://andrea.corbellini.name/2015/05/30/elliptic-curvecryptograpy-ecdh-and-ecdsa>. ECDH and ECDSA (2015)
3. Juels, A., Kosba, A., Shi, E.: The ring of gyges: investigating the future of criminal smart contracts. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS 2016, pp. 283–295. ACM (2016)
4. Urquhart, The inefficiency of bitcoin. *Econ. Lett.* **148**, 80–82 (2016)
5. Fairfield, J.: Smart contracts, bitcoin bots and consumer protection. *Washington Lee Law Rev. Online* **71**(2), 35–50 (2014)
6. Schuermann, F.: Bitcoin and beyond-a technical survey on decentralized digital currencies. *IEEE Commun. Surv. Tutorials*, 2084–2092 (2016)

7. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper **151**, 1–32 (2014)
8. Maxwell, G.: Coinjoin: A Bitcoin privacy for the real-world application, <https://bitcointalk.org/index.php?topic=279249.0> (2013)
9. Eyal, I., Gencer, A.E., Sirer, E.G., Van Renesse, R.: Bitcoin-ng: a scalable blockchain protocol. In: 13th fUSENIXg Symposium on Networked Systems Design and Implementation (fNSDIg 16), pp. 45–59 (2016)
10. Stark, Making sense of blockchain smart contracts. <http://www.coindesk.com/making-sense-smart-contracts/>. Accessed 16 July 2020
11. Samaniego, M., Deters, R.: Blockchain as a service for Iot. In: 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, pp. 433–436 (2016)
12. Feng, Q., He, D., Zeadally, S., Khan, M.K., Kumar, N.: A survey on privacy protection in blockchain system. *J. Netw. Comput. Appl.* **126**, 45–58 (2019)
13. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
14. Seijas, P.L., Thompson, S., McAdams, D.: Scripting smart contracts for distributed ledger technology. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10323 LNCS, pp. 631–632 (2017). <https://doi.org/10.1007/978-3-319-70278-0>
15. Morabito, V.: Smart contracts and licensing. In: *Business Innovation Through Blockchain*, pp. 101–124, Springer (2017)
16. Xu, X., et al.: The blockchain as a software connector. In: 2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA), pp. 182–191, IEEE (2016)
17. Werbach, K.D., Cornell, N.: Contracts ex machina. *Duke Law J.* **67**(2), 313–382 (2017)
18. Gilad, Y., Hemo, R., Micali, S., Vlachos, G., Zeldovich, N.: Al-gorand: Scaling byzantine agreements for cryptocurrencies. In: *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, pp. 51–68 (2017)
19. Verma, S.B., Saravanan, C.: Performance analysis of various fusion methods in multimodal biometric. In: 2018 International Conference on Computational and Characterization Techniques in Engineering & Sciences (CCTES), pp. 5–8. IEEE (2018)
20. Verma, S.B., Yadav, A.K.: Detection of hard exudates in retinopathy images. *ADCAIJ Adv. Distrib. Comput. Artif. Intell. J. Regular Issue* **8**(4), 41–48 eISSN: 2255–2863 (2019). <https://doi.org/10.14201/ADCAIJ2019844148>
21. Verma, S.B., Yadav, A.K.: Hard exudates detection: a review. In: Hassanien, A.E., Bhattacharyya, S., Chakrabati, S., Bhattacharya, A., Dutta, S. (eds.) *Emerging Technologies in Data Mining and Information Security*. AISC, vol. 1286, pp. 117–124. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-9927-9_12
22. Verma, S.B., Verma, S.B.: Data transmission in BPEL (business process execution language). *ADCAIJ Adv. Distrib. Comput. Artif. Intell. J. Regular* **9**(3), 105–117 (2020). eISSN: 2255–2863. <https://doi.org/10.14201/ADCAIJ202093105117>
23. Verma, S., et al.: Contactless palmprint verification system using 2-D Gabor filter and principal component analysis. *Int. Arab J. Inf. Technol.* **16**(1), 23–29 (2019)
24. Chandran, S., Verma, S.B.: Touchless palmprint verification using shock filter SIFT I-RANSAC and LPD IOSR. *J. Comput. Eng.* **17**(3), 2278–8727 (2015)
25. Verma, S.B., Pandey, B., Gupta, B.K.: Containerization and its architectures: a study. *ADCAIJ Adv. Distrib. Comput. Artif. Intell. J.* **11**(4), 395–409, eISSN: 2255–2863 (2022). <https://doi.org/10.14201/adcaij.28351>



A Quadratic Static Game Model for Assessing the Impact of Climate Change

Bouchra Mroué^{1(✉)}, Anthony Couthures¹, Samson Lasaulce^{1,2},
and Irinel Constantin Morărescu^{1,3}

¹ Université de Lorraine, CNRS, CRAN, 54000 Nancy, France
bouchra.mroue@univ-lorraine.fr

² Khalifa University, Abu Dhabi, UAE

³ Technical University of Cluj-Napoca, Cluj-Napoca, Romania

Abstract. The Conference of Parties (COP) aims to address the global warming problem through agreements for reducing emissions. However, the current strategies fall short due to minimal efforts for emission reductions driven by short-term economic considerations. To better understand under which circumstances countries may have more virtuous behavior, we propose a static game model. In this model, the players are represented by governments, and their actions correspond to emissions levels. The utilities are a trade-off between economic benefits and damage caused by climate change. A key feature of the game is that it is parameterized by a state, which is precisely the state of the climate dynamics. We conduct the Nash equilibrium analysis. In the numerical analysis, we assess the impact of the damage function on the behavior of the governments.

Keywords: Game theory · Climate change models · Complex systems · Static game · Potential game

1 Introduction

Global warming is a major environmental concern. The Conference of Parties (COP) was created to provide solutions to it. Nevertheless, one can see that efforts to emit fewer CO₂ are not very significant. On the contrary, the total of carbon emissions keeps on increasing whereas the effects of climate change have been made more apparent over the last decades. The objective of this paper is to provide some insights into the complex decision-making process of networked countries that optimize their utility functions by taking into account the dynamics of the global atmospheric temperature and CO₂ concentration. Doing so we provide elements that explain why the CO₂ emissions do not drastically reduce and exhibit some modelling conditions under which CO₂ reductions indeed occur.

The state of the art on this subject contains both geophysical-type and economic-type studies mainly based on empirical or ad-hoc strategies [15, 16].

Typical formal economic analyses do not integrate the geophysical aspect of the problem (see [1, 3] where the temperature dynamics are ignored). A neat game formulation that includes examining coalition formation, financial transfers, and cost-sharing was presented in [17]. This work provides insights into the complexity of environmental cooperation, coalition stability, and the design of efficient and stable agreements. Still, [17] only provides a game-theoretic analysis without considering the temperature but only CO₂ concentration dynamics. We also note that most of the existing game-theoretic studies are based on the work of Finus and his co-authors ([5–7]).

While the literature on climate change is quite rich, to our knowledge, no formal game-theoretic work has been conducted where both geophysical aspects and strategic aspects are considered and modeled mathematically. Models which couple economic aspects and climate science are referred to as integrated assessment models (IAMs). Among the most famous IAMs one can find the DICE model introduced by the Nobel Prize winner W. Nordhaus and his collaborators [14]. For convenience, they usually use simple climate model which matches the elaborate and complex geophysical models used by the IPCC (Intergovernmental Panel on Climate Change) [9].

In this study, we make the following key contributions to the field of environmental game theory:

- We propose a novel static game that is potential and where the utility function is a function of the geophysical state and the players' actions.
- We provide the expression of the unique pure Nash equilibrium, in some sufficient conditions, for the quadratic case functions.
- We assess numerically, the effects of the economic damage function due to climate change modeling on the behavior of the countries in terms of CO₂ emissions.

The rest of the paper is organized as follows. Section 2 is dedicated to the presentation of a simple but well-established climate model and its ingredients. The problem analyzed in this work is formulated and the subsequent game-theoretical analysis is provided in Sect. 3, where we study the existence and the uniqueness of the Nash equilibrium in a specific case. Numerical simulations illustrate our results in Sect. 4 and provide several insights or societal interest. We conclude the paper and give some perspectives in Sect. 5.

2 Simple Climate Model (SCM)

Let us first present the different pieces of a simple climate model that are coupled with the game introduced further in the paper. In our game, we are using the climate structure of IAMs involving three key ingredients: the carbon cycle (CC), the radiative forcing (RF), and the temperature dynamics (TD).

2.1 Carbon Cycle Model

In the sequel, we are using the CC model employed in [10] referred to as Joos model. This model was designed to fit the impulse response functions to a set of Earth System model simulations done by the CMIP5 model of IPCC. Denoting by C the vector of CO_2 concentrations in some boxes, where a box represents, in most of the carbon models, the proportion of carbon decaying with respect to the half-life of carbon in these virtual boxes. We get the following linear model:

$$\begin{aligned} C(t+1) &= \mathbf{A}_C C(t) + b_C E(t), \\ C_{\text{AT}}(t) &= d_C^T C(t). \end{aligned} \quad (1)$$

where

$$\mathbf{A}_C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.9975 & 0 & 0 \\ 0 & 0 & 0.9730 & 0 \\ 0 & 0 & 0 & 0.7927 \end{bmatrix}, \quad b_C = \begin{bmatrix} 0.2173 \\ 0.2240 \\ 0.2824 \\ 0.2763 \end{bmatrix}, \quad \text{and} \quad d_C = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

2.2 Radiative Forcing

The RF represents the impact of greenhouse gas (GHG) accumulation on the global radiation balance. The climate equation calculates the average surface temperature of the earth and the average deep-sea temperature at each time step. The relationship between the accumulation of GHGs and the increase in RF is derived from empirical measurements. A mathematical representation is given as:

$$F(t) = F_{2 \times \text{CO}_2} \log_2 \left(\frac{C_{\text{AT}}(t)}{C_{\text{AT,ref}}} \right) + F_{\text{nonCO}_2}(t), \quad \forall t \in \mathbb{R}^+, \quad (2)$$

where $C_{\text{AT,ref}}$ is the CO_2 concentration in 1750 that is considered as a reference since it is the pre-industrial time equilibrium, $F_{2 \times \text{CO}_2}$ is a parameter fitted from data and $F_{\text{nonCO}_2}(t)$ is the radiative forcing caused by other GHGs. In some models, such as FUND and PAGE, $F_{\text{nonCO}_2}(t)$ is modeled by the dynamics of methane and nitrous oxide while for DICE it represents exogenous forcing. In this work we use the exogenous forcings proposed by [11], given by:

$$F_{\text{nonCO}_2}(t) = f_0 + \min \left\{ f_1 - f_0, \frac{f_1 - f_0}{t_f} (t - 1) \right\}, \quad \forall t \in \mathbb{R}^+, \quad (3)$$

where f_0 and f_1 are respectively the forcing of GHGs other than CO_2 in 2010 and in 2100 and t_f is the time step.

2.3 Temperature Dynamic Model

The final part of an SCM is the Temperature Dynamics model. This allows us to describe the evolution of the global atmospheric temperature based on a specific

radiative forcing. The literature on the modeling of TD is very rich and we focus on the model proposed in [8] referred to as Geoffroy model:

$$\begin{cases} \theta(t + 1) = \mathbf{A}_\theta \theta(t) + b_\theta F(t + 1), \\ \theta_{\text{AT}}(t) = d_\theta^\top \theta(t). \end{cases} \tag{4}$$

where

$$\mathbf{A}_\theta = \begin{bmatrix} 1 - (\lambda + \mu)/c & \mu/c \\ \mu/c_0 & 1 - \mu/c_0 \end{bmatrix}, \quad b_\theta = \begin{bmatrix} 1/c \\ 0 \end{bmatrix}, \quad \text{and} \quad d_\theta = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{5}$$

We highlight that the dynamics depend on some constants: c as the effective heat capacity of the upper/mixed ocean layer, c_0 as the effective heat capacity of the oceans, λ and μ are chosen with all these constants such that they best fit the multi-model mean of the CMIP5 set which is the model used by IPCC [4]. In the sequel, we use the following values: $c = 7.3$, $c_0 = 106$, $\lambda = 1.13$, and $\mu = 0.73$. Denoting the state of the system by $x = (\theta, C)$, the atmospheric CO₂ concentration and the atmospheric temperature can be described as follows:

$$C_{\text{AT}}(t + 1) = C_{\text{AT}}(x(t), a(t)) = \psi_C(x(t)) + \tilde{b}_C \sum_{n=1}^N a_n(t), \tag{6a}$$

$$\theta_{\text{AT}}(t + 1) = \theta_{\text{AT}}(x(t), a(t)) = \psi_\theta(x(t)) + \tilde{b}_\theta \ln \left(\psi_C(x(t)) + \tilde{b}_C \sum_{n=1}^N a_n(t) \right), \tag{6b}$$

where $\tilde{b}_C = d_C^\top b_C$, $\tilde{b}_\theta = d_\theta^\top b_\theta F_{2 \times \text{CO}_2} / \ln 2$ are positive parameters and ψ_C, ψ_θ are functions of the state defined as $\psi_C(x(t)) = d_C^\top \mathbf{A}_C C(t)$ and

$$\psi_\theta(x(t)) = d_\theta^\top \mathbf{A}_\theta \theta(t) + d_\theta^\top F_{\text{nonCO}_2}(t) - d_\theta^\top b_\theta F_{2 \times \text{CO}_2} \log_2 C_{\text{AT,ref}}.$$

3 Game-Theoretic Analysis

3.1 Carbon Emission Game Model

This paper considers a static climate game Γ over a set of players $\mathcal{N} = \{1, \dots, N\}$ that represent non-identical countries. The goal of each player is to maximize their utility which is a trade-off between their benefits as functions of their emissions and a weighted global damage as a function of the global atmospheric temperature. The player’s action is the CO₂ emissions that they are planning to emit over the time step between two decisions (COP meetings). We note that e_n^{\min} and e_n^{\max} are respectively the minimum and the maximum emissions that player n can emit. The action set is $\mathcal{A} = \prod_{n \in \mathcal{N}} \mathcal{A}_n$ where $\mathcal{A}_n = [e_n^{\min}, e_n^{\max}] \subset \mathbb{R}$ is the set of actions for the player $n \in \mathcal{N}$. We shall also use a_{-n} to denote the vector of the actions of all the players except player n . The corresponding set of actions is \mathcal{A}_{-n} . We denote by $x \in \mathbb{R}^6$ the state of the system, by $a_n \in \mathcal{A}_n$ the action of player n , and by $a \in \mathcal{A}$ the vector of all actions. Note that the atmospheric temperature is a function of the state and of the action i.e., $\theta_{\text{AT}}(x, a)$.

Definition 1. *The utility function for the n^{th} player is chosen to be a difference between an individual benefit function B_n and a (weighted) global cost/damage function D :*

$$u_n(x, a) = \sum_{i=0}^2 \beta_{i,n} a_n^i - w_n \sum_{i=0}^2 \gamma_i \theta_{\text{AT}}^i(x, a) := B_n(a_n) - w_n D(\theta_{\text{AT}}(x, a)), \quad (7)$$

where w_n is a positive weight that measures the economic impact of climate change on player n . In the sequel we denote by $(u_n)_{n \in \mathcal{N}}$, the family of utility functions which defines the strategic form of the static game under study.

3.2 Existence and Uniqueness of a Pure Nash Equilibrium

A key solution concept for the interactive situation which involves several players, each aiming to maximize its own utility function, is given by the Nash equilibrium. A Nash equilibrium can be interpreted as a possible forecast for such a situation where decisions are interdependent as they are for the global carbon emission problem. An important property for a game is precisely to know whether it possesses a pure Nash equilibrium. It turns out that, by construction, the game under study has always a pure Nash equilibrium. This is because it belongs to the class of weighted potential games as defined by Monderer and Shapley [12].

A game $\Gamma = (\mathcal{N}, \mathcal{A}, \mathcal{U})$ is a weighted potential game if and only if there exists a potential function $\phi : \mathcal{A} \mapsto \mathbb{R}$ and $(\delta_n)_{n \in \mathcal{N}}$ a vector of positive weights, such that, for all $n \in \mathcal{N}$, $a_n, \tilde{a}_n \in \mathcal{A}_n; a_n \neq \tilde{a}_n$ and $a_{-n} \in \mathcal{A}_{-n}$ one has $u_n(a_n, a_{-n}) - u_n(\tilde{a}_n, a_{-n}) = \delta_n [\phi(a_n, a_{-n}) - \phi(\tilde{a}_n, a_{-n})]$.

It can be checked that the following function ϕ is a potential for the considered game with weights $(w_n)_{n \in \mathcal{N}}$:

$$\phi(x, a) = \sum_{n=1}^N \frac{1}{w_n} \sum_{i=0}^2 \beta_{i,n} a_n^i - \sum_{i=0}^2 \gamma_i \theta_{\text{AT}}^i(x, a). \quad (8)$$

The previous results yield the existence of at least one pure Nash equilibrium for the quadratic case. Next, we will provide a necessary condition for uniqueness. Actually, we consider a concave benefit function and a convex damage function, since the connection between GDP and emissions is frequently represented through a concave function. Also, a quadratic damage convex function is mostly used in the economic literature focusing on the consequences of climate change.

Proposition 1. *Supposing that $\gamma_2 > 0$ and for all $n \in \mathcal{N}$, if*

$$\frac{\gamma_2 \tilde{b}_\theta^2 \tilde{b}_C w_n}{e_n^{\max}} \exp\left(\frac{\gamma_1 + 2\gamma_2 \psi_\theta(x)}{2\gamma_2 \tilde{b}_\theta} - 1\right) - \frac{\beta_{1,n}}{2e_n^{\max}} < \beta_{2,n} < 0. \quad (9)$$

Then the pure Nash equilibrium is unique and it corresponds to all players emitting to the maximum, i.e., $a_n = e_n^{\max}$, $\forall n \in \mathcal{N}$.

Proof. If $\min_{a_n \in \mathcal{A}_n} B'_n(a_n)/w_n > \max_{a_n \in \mathcal{A}_n} \partial [D(\theta_{AT}(x, a))] / \partial a_n$, then one has a unique pure Nash equilibrium. Indeed, in this case, the utility functions will be strictly increasing and then the maximum is attained when all players emit the maximum of possible emissions. Straightforward computation shows that:

$$\min_{a_n \in \mathcal{A}_n} B'_n(a_n) = \min_{a_n \in \mathcal{A}_n} [\beta_{1,n} + 2\beta_{2,n}a_n] \stackrel{\text{if } \beta_{2,n} < 0}{=} \beta_{1,n} + 2\beta_{2,n}e_n^{\max}. \tag{10}$$

On the other hand:

$$\frac{\partial [D(\theta_{AT}(x, a))]}{\partial a_n} = \frac{\tilde{b}_\theta \tilde{b}_C \left[\gamma_1 + 2\gamma_2 \left[\psi_\theta(x) + \tilde{b}_\theta \ln \left(\psi_C(x) + \tilde{b}_C \sum_{n=1}^N a_n \right) \right] \right]}{\psi_C(x) + \tilde{b}_C \sum_{n=1}^N a_n}. \tag{11}$$

In this part of the proof, we will use the parameters $a, b, c, d, k \in \mathbb{R}$ that are constants, and independent of the problem formulated before. They are used to ease the presentation of the variation of the function. To find the maximum of (11), we consider the case where $\gamma_2 > 0$. Let us find the maximum of the function $f : \mathbb{R} \rightarrow \mathbb{R}$ for $a, b, c, k \in \mathbb{R}_+^*, d \in \mathbb{R}$, given for all $z \in \mathbb{R}$ by:

$$f(z) = \frac{k(d + c \ln(a + bz))}{a + bz}.$$

When differentiating f with respect to $z \in \mathbb{R}$, we find the unique root of f' given by $z_0 = (e^{(1-d/c)} - a) / b$. Then simply computing $f'((e^{-d/c} - a)/b) = kbc/e^{(4-2d/c)} > 0$, and $f'((e^{(2-d/c)} - a)/b) = -kbc/e^{(4-2d/c)} < 0$ provides that f' is strictly decreasing. We can conclude that the function f is strictly concave and reaches its maximum at z_0 , given by $f(z_0) = kc \exp(d/c - 1)$. Now by using f with $a = \psi_C(x) + \tilde{b}_C \sum_{m=1, m \neq n}^N a_m$, $b = \tilde{b}_C$, $c = 2\gamma_2 \tilde{b}_\theta$, $d = \gamma_1 + 2\gamma_2 \psi_\theta(x)$ and $k = \tilde{b}_\theta \tilde{b}_C$, we conclude that the maximum of D' is given by:

$$\max_{a_n \in \mathcal{A}_n} \frac{\partial [D(\theta_{AT}(x, a))]}{\partial a_n} = 2\gamma_2 \tilde{b}_\theta^2 \tilde{b}_C \exp \left(\frac{\gamma_1 + 2\gamma_2 \psi_\theta(x)}{2\gamma_2 \tilde{b}_\theta} - 1 \right).$$

After minimizing the benefit variations and maximizing the damage variations we get that $\min_{a_n \in \mathcal{A}_n} B'_n(a_n)/w_n > \max_{a_n \in \mathcal{A}_n} \partial [D(\theta_{AT}(x, a))] / \partial a_n$ is equivalent to:

$$\frac{\beta_{1,n} + 2\beta_{2,n}e_n^{\max}}{w_n} > 2\gamma_2 \tilde{b}_\theta^2 \tilde{b}_C \exp \left(\frac{\gamma_1 + 2\gamma_2 \psi_\theta(x)}{2\gamma_2 \tilde{b}_\theta} - 1 \right),$$

which is equivalent to (9). □

Proposition 1 basically states that looking at the short term, all the countries will emit as much as possible as long as the damage function does not have a sufficiently large impact. This can be changed either by considering less optimistic damage functions or looking at the long-term behavior when the atmospheric temperature is higher which will lead to larger damages.

3.3 Expression of the Nash Equilibrium

In this section, the goal is to express the Nash equilibrium actions for the player. The motivation for this is twofold; it makes interpretations much easier (e.g., the impact of radiative forcing or the damage severity level on the behavior of the countries) and it renders the problem of computing the equilibrium very simple to solve. To express the NE, let us assume from now on that ϕ is strictly concave. The pure NE is denoted by $a^* = (a_1^*, \dots, a_N^*)$ where either there exists $n \in \mathcal{N}$ such that $a_n^* \in \{e_n^{\min}, e_n^{\max}\}$, or a^* is an interior NE. In the later case, the players will tend to reduce their emissions.

We recall that the potential function is defined by (8) with the atmospheric temperature θ_{AT} given by:

$$\theta_{\text{AT}}(x, a) = \psi_{\theta}(x) + \tilde{b}_{\theta} \ln \left(\psi_C(x) + \tilde{b}_C \sum_{n=1}^N a_n \right). \quad (12)$$

Proposition 2. *If ϕ is strictly concave and differentiable, the Nash is the vector $a^* = (a_1^*, \dots, a_N^*)$ that satisfies, for all $n \in \mathcal{N}$, the following N equations*

$$\frac{1}{w_n} B'_n(a_n) - \frac{\partial [D(\theta_{\text{AT}}(x, a))]}{\partial a_n} = \bar{\lambda}_n - \underline{\lambda}_n. \quad (\text{KKT})$$

with $\bar{\lambda}_n, \underline{\lambda}_n \geq 0$ with $\underline{\lambda}_n^*(a_n^* - e_n^{\min}) = 0$ and $\bar{\lambda}_n^*(a_n^* - e_n^{\max}) = 0$ being the KKT multipliers with associated constraints.

Proof. The proof is straightforward: ϕ is continuous over \mathcal{A} , then there exists a NE, a^* . Moreover, if ϕ is strictly concave, then the NE is unique. Since the constraints are linear we can apply the KKT conditions. \square

In the proposition below, we provide sufficient conditions to express the NE. To do so, we will need the following lemma on the zeros of the equation $rs^2 + ps + q = k \ln(rs + v)$.

Lemma 1. *For $k, p, q \in \mathbb{R}$, $r, v \in \mathbb{R}_+^*$, and for all $s \in \mathbb{R}$ such that $rs + v > 0$, the following equation in s :*

$$rs^2 + ps + q = k \ln(rs + v),$$

- has at most one solution if $rk + (2v - p)^2/8 < 0$,
- has at most two solutions if $rk + (2v - p)^2/8 = 0$,
- has at most three solutions if $rk + (2v - p)^2/8 > 0$.

When ϕ is strictly concave, we can apply the KKT conditions to find the unique pure NE. Let us assume that ϕ is continuous on \mathcal{A} , so there exists a pure NE, denoted by a^* . Applying the Proposition 2, we have for every $n \in \mathcal{N}$, a^* verifies (KKT). This leads to the sufficient condition of the unique interior NE given in the following proposition.

Proposition 3. *Assuming that ϕ is strictly concave with $\beta_{2,n} \neq 0, \forall n \in \mathcal{N}$, and denoting $A_n := (w_n (\underline{\lambda}_n - \bar{\lambda}_n) - \beta_{1,n})/2\beta_{2,n}$. If*

$$\tilde{b}_\theta^2 \tilde{b}_C^2 \gamma_2 \sum_{n=1}^N \frac{w_n}{\beta_{2,n}} + \frac{(\psi_C(x) + \tilde{b}_C \sum_{n=1}^N A_n)^2}{8} < 0, \tag{13}$$

then the unique pure NE of the game Γ is given by $a^* = (a_1^*, \dots, a_N^*)$, where for all $n \in \mathcal{N}$,

$$a_n^* = A_n + \frac{w_n \tilde{b}_\theta \tilde{b}_C \left[\gamma_1 + 2\gamma_2 \psi_\theta(x) + 2\gamma_2 \tilde{b}_\theta \ln(\psi_C(x) + \tilde{b}_C \tilde{S}) \right]}{2\beta_{2,n} (\psi_C(x) + \tilde{b}_C \tilde{S})}, \tag{14}$$

where \tilde{S} is the unique solution of the equation $rs^2 + ps + q = k \ln(rs + v)$, with $r = \tilde{b}_C$, $p = \psi_C(x) - \tilde{b}_C \sum_{n=1}^N A_n$, $k = \tilde{b}_\theta^2 \tilde{b}_C \gamma_2 \sum_{n=1}^N \beta_{2,n}/w_n$, $v = \psi_C(x)$, and $q = -\psi_C(x) \sum_{n=1}^N A_n - \sum_{n=1}^N w_n \tilde{b}_\theta \tilde{b}_C (\gamma_1 + 2\gamma_2 \psi_\theta)/2\beta_{2,n}$.

Proof. Using Proposition 2, for all $n \in \mathcal{N}$ one has

$$\frac{1}{w_n} (\beta_{1,n} + 2\beta_{2,n} a_n^*) - [\gamma_1 + 2\gamma_2 \theta_{AT}(x, a^*)] \frac{\partial \theta_{AT}}{\partial a_n}(x, a^*) = \lambda_n - \bar{\lambda}_n.$$

Dividing by $\beta_{2,n} \neq 0$ and using the notation A_n introduced in the statement, we get that $\forall n \in \mathcal{N}$,

$$a_n^* = A_n + \frac{w_n \tilde{b}_\theta \tilde{b}_C \left[\gamma_1 + 2\gamma_2 \psi_\theta(x) + 2\gamma_2 \tilde{b}_\theta \ln(\psi_C(x) + \tilde{b}_C \sum_{n=1}^N a_n^*) \right]}{2\beta_{2,n} (\psi_C(x) + \tilde{b}_C \sum_{n=1}^N a_n^*)}. \tag{15}$$

Let us introduce the notation $s := \sum_{n=1}^N a_n^*$. Summing (15) over $n \in \mathcal{N}$ yields:

$$s = \sum_{n=1}^N A_n + \sum_{n=1}^N \frac{w_n}{2\beta_{2,n}} \frac{\tilde{b}_\theta \tilde{b}_C \left[\gamma_1 + 2\gamma_2 \psi_\theta(x) + 2\gamma_2 \tilde{b}_\theta \ln(\psi_C(x) + \tilde{b}_C s) \right]}{\psi_C(x) + \tilde{b}_C s},$$

which can be re-written as $rs^2 + ps + q = k \ln(rs + v)$ with r, p, q, v , and k given in the statement above. By using the results of the lemma 1, we conclude that if (13) is verified then there exists at most one solution \tilde{S} of the equation $rs^2 + ps + q = k \ln(rs + v)$ in $\left[\sum_{n=1}^N e_n^{\min}, \sum_{n=1}^N e_n^{\max} \right]$. Thus, the NE is unique.

4 Numerical Analysis

We will illustrate the previous theoretical results and the behavior of the proposed model. We have implemented a code that allows us to play with the parameters, whether they are from the SCM or the benefit and damage functions. We set $N = 6$ with the parameters specified in Table 1, where AOC refers

Table 1. Specific values for each player in 2020.

Player	China	USA	EU	India	Russia	AOC
e_n^{\max} (GtCO ₂ /y)	11	7	4	3	2	5
GDP _{<i>n</i>} ^{max} (10 ⁹ \$)	14630	19290	13890	2500	1420	11640
w_n	1.1847	1.1941	1.1248	0.9074	1.2866	1.1847

to all other countries. For the presented graphs, we will use the CC from [10] and the TD from [8], as they are often considered the closest to the IPCC results [4,13].

We illustrate the case where the benefit function is quadratic in a_n , and the damages are quadratic and re-scaled, i.e.,

$$u_n(x, a) = \text{GDP}_n^{\max} \left(2 \frac{a_n}{e_n^{\max}} - \left(\frac{a_n}{e_n^{\max}} \right)^2 - w_n [D(\theta_{AT}(x, a))]^\alpha \right), \quad (16)$$

where α represents the power of the damages and measures the severity level of climate change on the economics. The static game is played repetitively every five years until 2100 while updating e_n^{\max} and GDP_{*n*}^{max} at each iteration of the game. For more information, refer to [2].

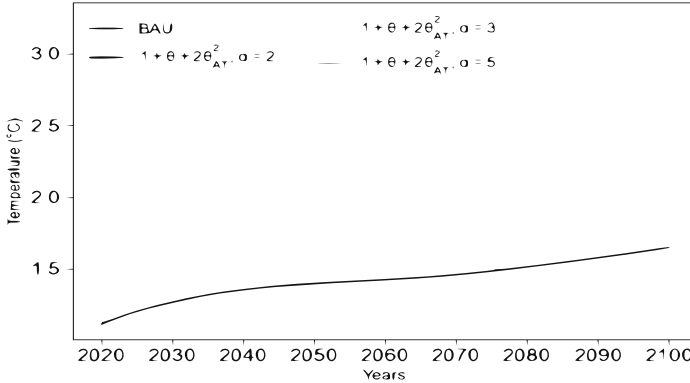


Fig. 1. The increase of the forecast temperature due to the CO₂emissions at NE in different scenarios.

It is interesting to note that higher α induces higher damages and consequently lower CO₂ emissions and smaller increases in the temperature. For large α (e.g., $\alpha = 5$), China, the USA, the EU, and AOC reduce their emissions until they completely stop emitting (see 2). In (Fig. 1), the temperatures in 2100 range from around +3.2°C for low damages, resembling a Business-as-Usual (BAU) scenario, to +1.6°C for high damages. These temperature levels are in line with

the projections of the IPCC [13] and correspond to the emission trajectories of the countries. In order to prevent the over-warming of the planet by 2100 we need to revise the modeling of the economic damages and change the strategies accordingly.

Table 2. Time at which the countries stop emitting versus α (which measures the economic damage due to climate change). The symbol / means no stopping.

Player	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$	$\alpha = 6$	$\alpha = 7$	$\alpha = 8$	$\alpha = 9$	$\alpha = 10$
China	/	/	2065	2020	2020	2020	2020	2020	2020	2020
USA	/	/	2075	2020	2020	2020	2020	2020	2020	2020
EU	/	/	/	2045	2020	2020	2020	2020	2020	2020
India	/	/	/	/	/	/	/	2095	2085	2080
Russia	/	/	/	/	/	/	/	/	/	/
AOC	/	/	2080	2025	2020	2020	2020	2020	2020	2020

Table 2 shows that if the damages are not significant, i.e. α is small, the CO₂ emissions of the players will not stop before 2100. Low damage hampers the cooperation recommended by the IPCC. When α is large enough, the NE strategies of the players are to stop emitting as soon as possible. Except for Russia which continues to emit no matter how big is the damage, and for India which stops emitting only when $\alpha \geq 8$. This can be explained by the fact that the benefits of India and Russia are still very big compared with the corresponding loss. The product $w_n \text{GDP}_n^{\max}$ has to be increased for these countries in order to stop their emissions.

5 Conclusion

In this paper, we have introduced and analyzed a static game that provides some insights into the strategic behavior of governments in terms of carbon emissions with respect to climate change. Each government is assumed to implement a trade-off between a benefit due to emitting and damage due to climate change. The Nash equilibrium of the game being a suitable solution concept, we have studied the existence and uniqueness of the equilibrium and also addressed the problem of expression of the actions at equilibrium. The numerical analysis provides several insights into the carbon emission problem. For instance, it is seen that to reach the Paris Agreement on climate (namely, maintain the temperature excess below 2°C), the damage to climate change has to be significant enough. This constitutes a sufficient condition under which governments will spontaneously reduce their emissions. Depending on the severity level of the damage (which is measured by the exponent α), governments are incited to stop emitting CO₂ and it is shown to be possible to (roughly) forecast a time at which a country stops emitting. The obtained times are typically higher than values

claimed publicly (e.g., 2050). To conclude this paper, we would like to mention several extensions of the present work. First, the present problem formulation might be enriched by considering the planning aspect for which countries consider long-term utilities instead of short-term ones. Second, the emergence of cooperation might be studied by considering other solution concepts such as the social optimum or Nash bargaining solution. Thirdly, the damage functions might be more individualized while maintaining the potential structure of the game. At last, the present work can be seen as the first necessary step to be taken to study formally repeated interactions between the players, e.g., through a repeated or stochastic game model. In such a framework, it might be assumed that the action and state of a country are not perfectly observed by the others, which defines a non-trivial observation graph to be taken into account in the equilibrium analysis.

References

1. Al Khouradjie, A., Finus, M.: Measures to enhance the effectiveness of international climate agreements: the case of border carbon adjustments. *Eur. Econ. Rev.* **124**, 103405 (2020)
2. Anthony, C., Bouchra, M.: Co2-game (2023). <https://doi.org/10.5281/zenodo.8283364>
3. Bosetti, V., et al.: Incentives and stability of international climate coalitions: an integrated assessment. *Energy Policy* **55**, 44–56 (2013)
4. Dietz, S., et al.: Are economists getting climate dynamics right and does it matter. *J. Assoc. Environ. Resour. Economists* **8**, 895–921 (2021)
5. Finus, M.: Game theoretic research on the design of international environmental agreements: insights, critical remarks, and future challenges. *Int. Rev. Environ. Resour. Econ.* **2**, 29–67 (2008)
6. Finus, M.: *Game Theory and International Environmental Cooperation: Any Practical Application?* Edward Elgar Publishing (2000)
7. Finus, M., et al.: The efficacy of international environmental agreements when adaptation matters: Nash-Cournot vs Stackelberg leadership. *J. Environ. Econ. Manage.* **109**, 102461 (2021)
8. Geoffroy, O., et al.: Transient climate response in a two-layer energy-balance model. Part I: analytical solution and parameter calibration using CMIP5 AOGCM experiments. *J. Clim.* **26**, 1841–1857 (2013)
9. IPCC, Annex II: Models [Gutiérrez, J M., A.-M. Tréguier (eds.)], in: Masson-Delmotte, V., et al. (Eds.), *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change* (2021)
10. Joos, F., et al.: Carbon dioxide and climate impulse response functions for the computation of greenhouse gas metrics: a multi-model analysis. *Atmos. Chem. Phys.* **13**, 2793–2825 (2013)
11. Kellett, C.M., et al.: Feedback, dynamics, and optimal control in climate economics. *Annu. Rev. Control* **47**, 7–20 (2019)
12. Monderer, D., Shapley, L.S.: Potential games. *Games Econ. Behav.* **14**, 124–143 (1996)

13. Mukherji, A., et al.: Synthesis report of the IPCC sixth assessment report (AR6) (2023)
14. Nordhaus, W., Sztorc, P.: DICE 2013R : Introduction and User's Manual (2013). URL <https://sites.google.com/site/williamdnordhaus/dice-rice>
15. Perissi, I., Jones, A.: Investigating EU decarbonization strategies: evaluating the pathway to carbon neutrality by 2050. *Sustainability* **14**, 4728 (2022)
16. Solano, B., et al.: Decarbonizing the EU energy system by 2050: an important role for BECCS. *Clim. Policy* **17**, S93–S110 (2017)
17. Tulkens, H.: *Economics, Game Theory and International Environmental Agreements: The Ca' Foscari Lectures*. World Scientific Publishing (2019)



Linear Stochastic Processes on Networks and Low Rank Graph Limits

Alex Dunyak^(✉) and Peter E. Caines

Department of Electrical and Computer Engineering, McGill University, Montreal,
QC H3A 0E9, Canada

alex.dunyak@mail.mcgill.ca, peterc@cim.mcgill.ca

Abstract. The modelling of stochastic linear systems in large complex networks is intractable computationally and may be impossible due to data-collection costs or privacy concerns. Graphon theory provides an approach to overcome these issues by providing potentially simple limit objects for infinite sequences of graphs, permitting one to approximate arbitrarily large networks by infinite dimensional operators. Graphon system theory is extended here to stochastic systems by the use of Q -noise, a generalization of Wiener processes in finite dimensional spaces to processes in function spaces. The theory is developed for low rank systems as a special case.

Keywords: Random graphs · Graphons · Low rank approximations

1 Introduction

Large graphs are common objects in modern society. From the Internet of Things to electrical generation and distribution to social networks, complex networks are the focus of intense research. Moreover, dynamic systems on networks constitute the models for many phenomena which have a significant impact on society, for instance epidemics [6], unemployment rates [5], and consensus formation [7]. However, for sufficiently large networks such problems are intractable when tackled with standard analysis methods.

One approach to modelling large networks is to approximate a convergent graph sequence by a function called a graphon [22]. Informally, a graphon is the limiting function of an adjacency matrix mapped to the unit square. Using graphons for modelling large graphs replaces very large complex networks with a single function, which may be useful in the modelling and design of complex systems.

Much of the previous work with dynamic graphon systems has been focused on deterministic control systems ([12, 13]), and stochastic mean field games on graphons have been investigated in ([3, 4, 15, 24]). A centrality measure for

*A. Dunyak, **P.E. Caines: Work supported by NSERC 2019-05336, ARL W911NF1910110, Air Force OSR Grant FA9550-23-1-0015.

graphon mean field games is provided in [11]. Gao and Caines [14] showed that deterministic harmonic oscillators can be optimally controlled by a low rank approximation of the network, and that the error between the graphon limit and the finite network can be made arbitrarily small. Dunyak and Caines [9] showed that Gaussian noise on the unit interval ([10, 16, 21]), termed Q-noise, is the appropriate limit for system noise on sequences of systems subject to Brownian disturbances, as would arise, for instance, in stochastic versions of the deterministic graphon systems above. Independently, a numerical approach for systems of this type was explored in [23]. In this article it is numerically illustrated that the behavior of stochastic systems on large networks can be adequately modelled by their graphon limits. In particular, when the limit graphon is low rank, the limit model can be efficiently computed with a complexity depending only on the rank.

1.1 Motivation: Networked Systems and Graphons

Define a graph $G_A^N = (V_N, E_A^N)$ with $N < \infty$ vertices, with associated adjacency matrix A^N . Let $x^N : [0, T]^N \rightarrow \mathbb{R}$ be a vector of states where the i th value is associated with the state of the i th vertex of the graph. For clarity of notation, systems where each node has a single state are considered. The theory can be extended to systems where each node has multiple states as well. Let the (i, j) th entry of the matrices A^N be the impact of the state at node i on node j , respectively. For each node, define a Brownian motion W_i^N disturbance with positive covariance matrix Q^N . Let a_N be a constant describing the impact of the state of a node on itself.

Finally define a networked system on a graph with the following equation for each node,

$$d[x_t^i]_{i=1}^N = \left(\frac{1}{N} \sum_{j=1}^N A_{ij}^N x_t^j + a_N x_t^i \right) dt + dW_i^N(t) =: dx_t^N. \tag{1}$$

As the graph becomes larger, the networked system adjacency matrix A^N approaches its associated graphon [22], which is a bounded measurable function mapping $[0, 1] \times [0, 1] \rightarrow [0, 1]$, denoted \mathbf{A} . The graphon limit system is denoted as

$$d\mathbf{x}_t = (\mathbf{A} + a\mathbb{I})\mathbf{x}_t dt + d\mathbf{w}_t, \tag{2}$$

where \mathbf{x}_t is a square-integrable function on the unit interval, \mathbf{A} is a graphon, a is a real constant, \mathbb{I} is the identity operator, and \mathbf{w}_t is a Q-noise, a generalization of Gaussian noise from finite-dimensional vectors to the unit interval which will be explained in detail in Sect. 2.2. An example of this network convergence is shown with the two finite networks in Fig. 1 converging to the graphon limit in Fig. 2. In particular, this shows a uniform attachment graph [22] converging to its graphon. If the adjacency matrix of a graph sequence converges to a continuous graphon $\mathbf{W}(\alpha, \beta)$ in an ordered manner, the normalized degree of a node associated

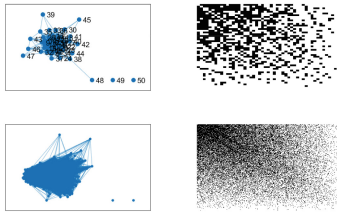


Fig. 1. Graphs with 50 and 500 nodes, respectively, where the “pixel picture” of the associated adjacency matrices converge to the graphon in Fig. 2 when mapped to the unit square. Lower indexed nodes are more likely to be connected than higher indexed nodes.

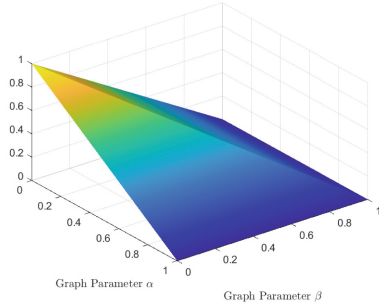


Fig. 2. The graph sequence converges to the uniform attachment graphon $W = (\alpha, \beta) = 1 - \max(\alpha, \beta)$, $\alpha, \beta \in [0, 1]$.

with parameter α approaches the integral of $W(\alpha, \beta)$ over β . In this sense, the fraction of vertices in a large graph that are neighbors with a specific node with parameter α is given by the graphon, as shown in Fig. 3. The absolute value of the eigenvalues are shown in Fig. 4. As there is a single eigenvalue with a much larger magnitude than the others, this implies that there is one eigenvector of the adjacency matrix that dominates all other eigenvectors.

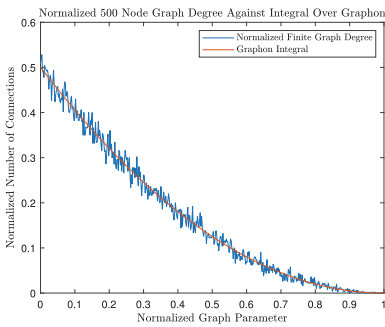


Fig. 3. The normalized degree of the 500 node adjacency matrix in Fig. 1 plotted against the integral of $W(\alpha, \beta) = 1 - \max(\alpha, \beta)$ over $\beta \in [0, 1]$.

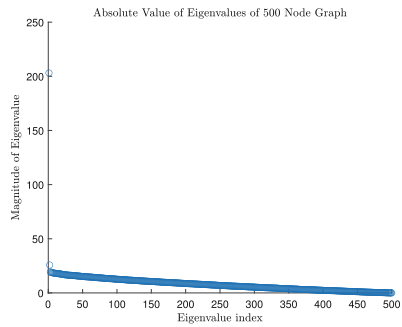


Fig. 4. The distribution of the absolute value of eigenvalues in the 500 node adjacency matrix. The largest eigenvalue has a magnitude of 202.7 (with index 0), the second (with index 1) has a magnitude of 26.0, and the rest of the eigenvalue magnitudes lie below 20, clustering at zero.

Another classic example is the Erdős-Renyi graph (also known as a Poisson graph [19]), where each node in the graph is connected to other nodes with probability p , as in Fig. 5. The resulting graph is a dense network with a structure that is complex for any finite network size, but is very simple in the limit (Fig. 6).

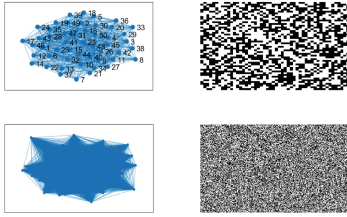


Fig. 5. Graphs with 50 and 500 nodes, respectively, of an Erdős-Rényi graph with connection probability $p = 0.5$. As the number of nodes increases, the adjacency matrix converges analytically to the constant $\mathbf{6}$.

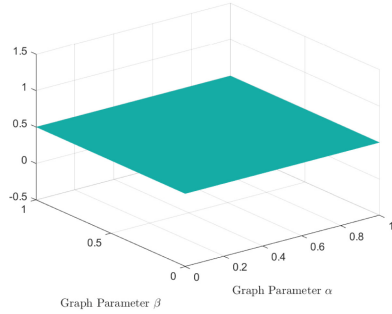


Fig. 6. The graph sequence in Fig. 5 converges to the constant graphon $\mathbf{W}(\alpha, \beta) = 0.5, \alpha, \beta \in [0, 1]$.

W-random graphs [22], an extension of Erdős-Rényi graphs, determine the connection probability between two nodes by assigning a latent parameter $\alpha \in [0, 1]$ to $\mathbf{W} : [0, 1] \times [0, 1] \rightarrow [0, 1]$. As the number of nodes in the graphs increases, the adjacency matrix can be mapped to the kernel \mathbf{W} . This approach uses the geometric intuition behind graphs defined on latent spaces (e.g. [20]), in this case the nodes are uniformly distributed on the space $[0, 1]$.

Stochastic block matrices [18] can be considered a special case of W-random graphs, where the relevant kernel is a piece-wise constant function on the unit square.

2 Preliminaries

2.1 Notation

- The set of vectors of real numbers of dimension m is denoted \mathcal{R}^m .
- Graphons (i.e. bounded symmetric $[0, 1]^2$ functions used as the kernels of linear integral operators) are denoted in italicized bold capital letters, such as \mathbf{A} .
- $L_2[0, 1]$ denotes the Hilbert space of real square-integrable functions on the unit interval which is equipped with the standard inner product.
- The $L_2[0, 1]$ identity operator is denoted \mathbb{I} , such that $\mathbb{I}\mathbf{u} = \mathbf{u}$ for all $\mathbf{u} \in L_2[0, 1]$. In finite dimensions, the \mathcal{R}^n identity operator is denoted I .
- Operators of the font \mathbb{A} have the structure $\mathbb{A} = \mathbf{A} + a\mathbb{I}$, where \mathbf{A} is a graphon and a is a real scalar.
- A linear integral operator with the kernel $\mathbf{Q} : [0, 1]^2 \rightarrow \mathcal{R}$ acting on a function $\mathbf{f} \in \mathcal{L}^2[0, 1]$ is defined by

$$(\mathbf{Q}\mathbf{f})(x) = \int_0^1 \mathbf{Q}(x, y)\mathbf{f}(y)dy, \quad \forall x \in [0, 1]. \tag{3}$$

When unambiguous, the argument x of $(\mathbf{Q}\mathbf{f})(x)$ is dropped.

- Denote \mathcal{Q} to be the set of bounded symmetric non-negative functions. A symmetric function $\mathbf{Q} : [0, 1]^2 \rightarrow \mathcal{R}$ is non-negative if the following inequality is satisfied for every function $\mathbf{f} \in \mathcal{L}^2[0, 1]$,

$$0 \leq \int_0^1 \int_0^1 \mathbf{Q}(x, y) \mathbf{f}^*(x) \mathbf{f}(y) dx dy := \langle \mathbf{Q} \mathbf{f}, \mathbf{f} \rangle < \infty. \tag{4}$$

- A partition of the unit interval of N increments is denoted $P^N = \{P_1, \dots, P_N\}$, where $P_1 = [0, \frac{1}{N}]$ and $P_i = (\frac{i-1}{N}, \frac{i}{N}]$. An $L_2[0, 1]$ function which is piece-wise constant on each section P_k of the unit interval is denoted $\mathbf{v}^{[N]}$, and a self-adjoint $L_2[0, 1]$ operator \mathbf{M} which is piece-wise constant on the Cartesian product $P^N \times P^N$ is denoted $\mathbf{M}^{[N]}$ (or $\mathbb{M}^{[N]}$, if it is of the form $\mathbb{M}^{[N]} = \mathbf{M}^{[N]} + m\mathbb{I}$). This formulation is necessary for mapping N dimensional graph systems to the unit interval, as in Sect. 2.4.

2.2 Q-Noise

Q-noise processes, first applied to graphon systems independently in [9, 23], are $L_2[0, 1]$ valued random processes that satisfy the following axioms.

1. Let $\mathbf{w}(\alpha, t, \omega) : [0, 1] \times [0, T] \times \Omega \rightarrow \mathcal{R}$ for all $t \in [0, T]$, $\alpha \in [0, 1]$, $\omega \in \Omega$ be a random variable where, for each $t \in [0, T]$, $\mathbf{w}(t, \alpha, \omega)$ is square-integrable in α . For notation, ω is suppressed when the meaning is clear.
2. For all $\alpha \in [0, 1]$, $\mathbf{w}(\alpha, t) - \mathbf{w}(\alpha, s)$ is a Brownian motion increment in time for all $t, s \in [0, T]$, with $\mathbf{w}(\alpha, t) - \mathbf{w}(\alpha, s) \sim \mathcal{N}(0, |t-s|\mathbf{Q}(\alpha, \alpha))$ where $\mathbf{w}(\alpha, 0) = 0$ for all $\alpha \in [0, 1]$.
3. Let $\mathbf{w}_{t-t'}(\alpha) = \mathbf{w}(\alpha, t) - \mathbf{w}(\alpha, t')$. Then $\mathbb{E}[\mathbf{w}_{t-t'}(\alpha) \mathbf{w}_{s-s'}(\beta)] = |[t, t'] \cap [s, s']| \cdot \mathbf{Q}(\alpha, \beta)$.
4. $\mathbf{w}(\alpha, t, \omega) - \mathbf{w}(\beta, s, \omega)$ is piece-wise continuous almost everywhere in t, s and α, β .

Q-noise processes have a natural construction in separable Hilbert spaces. Let $\{W_1, W_2, \dots\}$ be a sequence of independent Brownian motions. Let $\mathbf{Q} \in \mathcal{Q}$ have a diagonalizing orthonormal basis $\{\phi_k\}_{k=1}^\infty$ with eigenvalues $\{\lambda_k\}_{k=1}^\infty$. Then

$$g(\alpha, t, \omega) = \sum_{k=1}^\infty \sqrt{\lambda_k} \phi_k(\alpha) W_k(t, \omega) \tag{5}$$

is a Q-noise process lying within the span of the processes $\{W_1, W_2, \dots\}$. By Mercer’s theorem (see, e.g. [17]), \mathbf{Q} has the eigenvalue and basis representation:

$$\mathbf{Q}(x, y)(t) = \sum_{r=1}^\infty \sqrt{\lambda_r} \phi_r(x) \phi_r(y) = \mathbb{E}[g(\alpha, t, \omega) g(\beta, t, \omega)]. \tag{6}$$

The common name for such a process is Q-Wiener process ([10, 16]). A formulation like this is necessary, as when the covariance operator \mathbf{Q} is unbounded, the random variable \mathbf{w}_t may have a singular probability measure.

2.3 Linear Dynamical Systems

Definition 1 (Q-noise Dynamical Systems). Let $\mathbf{x} : [0, 1] \times [0, T] \rightarrow \mathcal{R}$ be an $\mathcal{L}^2[0, 1] \times [0, T]$ function with a given initial condition $\mathbf{x}(\cdot, 0) = \mathbf{x}_0$. Let $\mathbb{A} \in \mathcal{M}$ be a bounded linear operator from $\mathcal{L}^2[0, 1]$ to $\mathcal{L}^2[0, 1]$ such that $\mathbb{A}\mathbf{Q}\mathbb{A}^* \in \mathcal{Q}$. This defines a Q-noise denoted \mathbf{w}_t .

Given $\mathbf{x}_0 \in L_2[0, 1]$, a linear dynamical system with Q-noise is an infinite dimensional differential system satisfying the following equation,

$$d\mathbf{x}_t(\alpha) = (\mathbb{A}\mathbf{x}_t)(\alpha)dt + d\mathbf{w}(\alpha, t), \tag{7}$$

where, for a partition of $[0, t]$, $(0, t_2, \dots, t_{N-2}, t)$,

$$\int_0^t d\mathbf{w}(\alpha, s) = \lim_{N \rightarrow \infty} \sum_{k=1}^N (\mathbf{w}(\alpha, t_{k+1}) - \mathbf{w}(\alpha, t_k)), \tag{8}$$

in the mean-squared convergence sense.

2.4 Finite Network Systems

Consider a networked system of the form,

$$dx_t = \left(\frac{1}{N}(A^N x_t) + a_N x_t\right)dt + dW_t^N, \tag{9}$$

where A^N is an $N \times N$ adjacency matrix and W_t^N is a collection of N Wiener processes with covariance matrix Q^N . The finite dimensional system is mapped to piecewise constant functions on the unit square (see [13]). Define the uniform partition on the unit interval as $P^N = \{P_1, \dots, P_N\}$, where $P_1 = [0, \frac{1}{N}]$ and $P_i = (\frac{i-1}{N}, \frac{i}{N}]$. Then, the following step function graphon for N nodes can be defined for all $x, y \in [0, 1]$:

$$\mathbf{A}^{[N]}(x, y) = \sum_{i=1}^N \sum_{j=1}^N A_{ij}^N \mathbb{1}_{P_i}(x) \mathbb{1}_{P_j}(y). \tag{10}$$

Define the covariance of the disturbance as a piece-wise constant function $\mathbf{Q}^{[N]}$ analogous to the finite dimensional covariance matrix in a similar manner. By mapping the Wiener processes W_t^N to the corresponding elements of the unit interval $\mathbf{w}_t^{[N]}$ the corresponding system in $\mathcal{L}^2[0, 1]$ can be expressed as

$$d\mathbf{x}_t^{[N]} = (\mathbf{A}^{[N]} + a_N \mathbb{I})\mathbf{x}_t^{[N]} dt + d\mathbf{w}_t^{[N]}. \tag{11}$$

This system, defined on piece-wise constant functions on the unit interval, is identical to the system on the network. Further, as the network size grows infinitely large and if $\mathbb{A}^{[N]}$ converges to \mathbb{A} in the L_2 operator norm, the network system solution converges to the solution of the graphon system in the $L_2[0, 1]$ norm,

$$d\mathbf{x}_t = (\mathbf{A} + a\mathbb{I})\mathbf{x}_t dt + d\mathbf{w}_t^\infty. \tag{12}$$

The proof is presented in [9].

3 Numerical Examples

For the following numerical simulations, the unit interval $[0, 1]$ is partitioned into N segments. In each example, $N = 300$, and the state of the simulated systems follow the form

$$dx_t^{[N]} = (\mathbf{A} + a\mathbb{I})\mathbf{x}_t^{[N]}dt + d\mathbf{w}_t^{[N]}, \quad (13)$$

$$\mathbf{x}_0^{[N]}(\alpha) = \sin(2\pi\alpha), \quad (14)$$

as in Eq. (11). This discretized system is used as an approximate solution to the infinite dimensional system. A terminal time of $T = 1$ and Euler's method with a time increment of $\Delta t = 0.001$ are used.

3.1 Low Rank Graphon Systems

Approximation of Large Random Graph System with Finite Rank Limit. One of the primary areas of interest in graphon theory is to approximate systems on large graphs with their corresponding graphon systems. One approach to creating these random graphs is to use the W-random graph method. This method takes a set of vertices V , where each vertex has a type $\alpha_i \in [0, 1]$, a kernel $\mathbf{M} : [0, 1]^2 \rightarrow [0, 1]$ and, for each pair v_i and v_j , creates an edge e_{ij} with probability $\mathbf{M}(\alpha_i, \alpha_j)$ independently. To demonstrate that a low rank graphon system is computationally simpler than a finite graph system, consider the basis function \mathbf{f} and kernel \mathbf{M} ,

$$\mathbf{f}(\alpha) = \alpha^2 - 1, \quad \alpha \in [0, 1], \quad (15)$$

$$\mathbf{M}(\alpha, \beta) = \mathbf{f}(\alpha)\mathbf{f}(\beta), \quad \alpha \in [0, 1], \beta \in [0, 1], \quad (16)$$

where the features α_i are each mapped uniformly to the unit interval, with a spacing of $\frac{1}{300}$ units. The finite graph system is approximated by finding a 300 node graph using the W-random graph method with kernel \mathbf{M} , and has adjacency matrix A^N . This is mapped to the piecewise constant graphon $\mathbf{A}^{[N]}$. The adjacency matrix used for this demonstration is shown in Fig. 7, which has the graphon limit Fig. 8.

The Q-noise covariance \mathbf{Q} is set to be $\mathbf{Q} = 2\mathbf{M}$. Then, the two systems under consideration are the piecewise constant graphon system defined by $\{\mathbb{A}^{[N]} = \mathbf{A}^{[N]} + \mathbb{I}, \mathbf{Q}\}$ and the limit graphon system $\{\mathbb{A} = \mathbf{M} + \mathbb{I}, \mathbf{Q}\}$, the initial condition is set to be $\mathbf{x}_0^{[N]}(\alpha) = \mathbf{x}_0(\alpha) := \sin(2\pi\alpha)$, $\alpha \in [0, 1]$, and their trajectories are shown in Fig. 9. While the adjacency matrix A^N has some structure, it is nearly full rank despite being generated from a low rank function. Thus, characterizing the finite graph system state x_t^N defined by the parameters $\{\mathbb{A}^{[N]}, \mathbf{Q}\}$ requires a state space of 300 states, one state for each node. Further, adding nodes to the graph increases the computational complexity of the system quadratically, as each additional node increases the number of connections in the graph. The limit graphon system \mathbf{x}_t defined by the parameters $\{\mathbb{A}, \mathbf{Q}\}$, is effectively rank two, despite existing in $L_2[0, 1]$. The system has two modes,

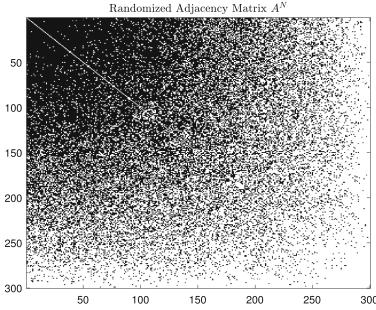


Fig. 7. The adjacency matrix found using the W -graph generation method with kernel $M(\alpha, \beta) = f(\alpha)f(\beta)$. Notably, the adjacency matrix is rank 299.

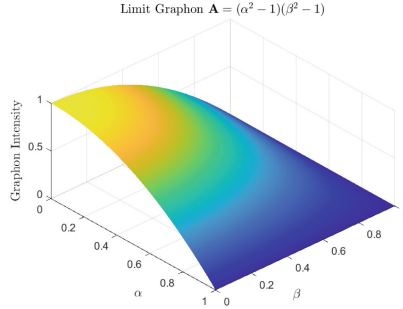


Fig. 8. The graph limit of A^N , which converges to the graphon $A(\alpha, \beta) = (\alpha^2 - 1)(\beta^2 - 1)$. This is a single dimensional linear operator, mapping any function $x \in L^2[0, 1]$ to the space spanned by $f(\alpha) = (\alpha^2 - 1)$, $\alpha \in [0, 1]$.

given by the single eigenfunction of A and Q (assumed to have identical invariant subspaces) and the orthogonal complement of the initial condition with the eigenfunction denoted \check{x}_0 . Recalling that $\langle f, \phi \rangle \phi$ is the orthogonal projection of f onto a normalized function ϕ , the state trajectory can be described with the following system of differential equations,

$$\mathbf{x}_t = x_t^\phi \cdot \phi + \check{\mathbf{x}}_t, \quad x_t^\phi \in \mathcal{R}, \quad \check{\mathbf{x}}_t \in L_2[0, 1] \tag{17}$$

$$\phi(\alpha) = \frac{\alpha^2 - 1}{\sqrt{\int_0^1 (\beta^2 - 1)^2 d\beta}}, \tag{18}$$

$$dx_t^\phi = (\langle A\phi, \phi \rangle + a)x_t^\phi dt + d\langle \mathbf{w}_t, \phi \rangle, \tag{19}$$

$$x_0^\phi := \langle \mathbf{x}_0, \phi \rangle = \langle \sin(2\pi\alpha), \phi \rangle, \tag{20}$$

$$d\check{\mathbf{x}}_t(\alpha) = a\check{\mathbf{x}}_t(\alpha)dt, \tag{21}$$

$$\check{\mathbf{x}}_0(\alpha) = \sin(2\pi\alpha) - \langle \phi, \mathbf{x}_0 \rangle \phi(\alpha). \tag{22}$$

Notably, as the covariance operator Q of \mathbf{w}_t shares an eigenfunction with the driving operator A , the Q -noise \mathbf{w}_t is one dimensional, and only impacts the state along the ϕ dimension. Hence, the orthogonal component $\check{\mathbf{x}}_t(\alpha)$ is deterministic and, for each $\alpha \in [0, 1]$, evolves independently of all other indices. The system evolution is shown in Fig. 10. The root mean squared error over the time interval $[0, 1]$, shown in Fig. 13, indicates that the rank one graphon closely approximates the finite graph system.

3.2 Stochastic Block Matrices

Stochastic block matrices are special cases of W -random graphs, as the kernel used is a piece-wise constant function. As a result, every system defined on a

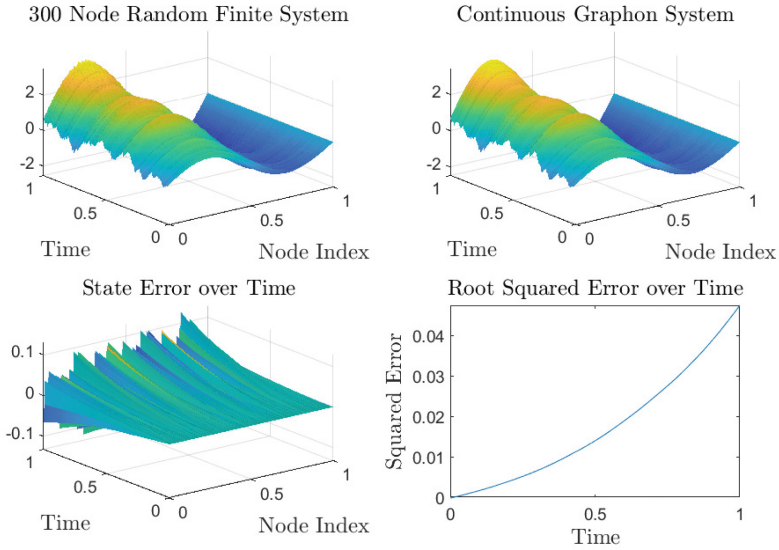


Fig. 9. Top Left: the state trajectory evolved using 7 as the driving operator. Top Right: Graphon approximation of the system. Bottom Left: The error between the original system and the graphon limit system. Bottom Right: The root of the squared error of the system state over time.

stochastic block matrix is finite rank, with eigenfunctions corresponding to the eigenvectors of the kernel matrix.

For example, suppose that there is a system on a graph where each node is contained in one of two sets, S_1 and S_2 , and that there are 150 nodes of each set. A node in set S_1 connects to a node of set S_1 with probability 0.8, and to a node of set S_2 with probability 0.2. Nodes of set S_2 connect to other S_2 nodes with probability 0.5. The adjacency matrix of such a network is shown in Fig. 14, and the corresponding limit graphon kernel is shown in Fig. 15. The system on the associated finite graph is shown in Fig. 16, and the graphon limit system is shown in Fig. 17. The two trajectories are very similar, indicating that the rank two graphon system is a good approximation.

4 Future Directions

The primary future directions for this research are in the applications of low rank graphon theory and Q-noise theory to various systems. For instance, an SIR epidemic model of a population on a very large graph can be modelled as a system on a graphon [8], and the use of low rank graphons gives insight into the evolution of such systems. A similar approach has been used to model opinion dynamics on piece-wise constant graphons (i.e. graphons associated to stochastic block matrices), a special case of low rank graphons [1]. Currently, the introduction of Q-noise into the deterministic estimation and control framework of [12, 13], and in particular to the low rank analysis of [14], is under development.

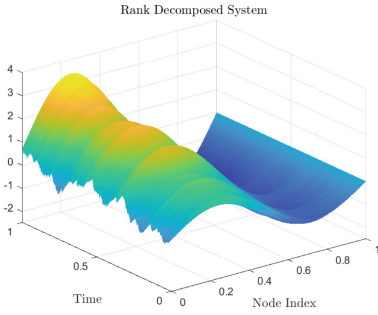


Fig. 10. The low rank approximation of the adjacency matrix system shown in Fig. 9. It is additively composed of two components, shown in Figs. 11 and 12.

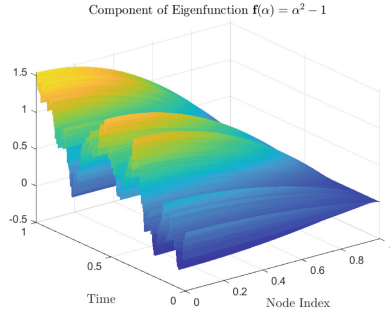


Fig. 11. The state component of the initial condition projected onto the eigenfunction of A . In this case, the normalized eigenfunction is $\phi(\alpha)$, shown above.

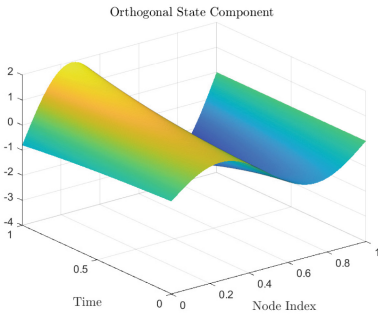


Fig. 12. The component of the initial condition orthogonal to $\phi(\alpha)$. Because the covariance operator of the noise is rank one and shares the same eigenfunction as A , there is no noise in the orthogonal component.

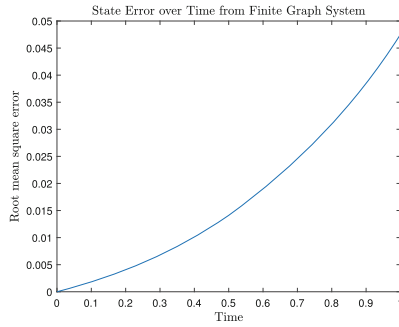


Fig. 13. The root mean squared error as the system evolves over time. The error at $T = 1$ is less than 0.05, indicating that the 300 node system is well approximated by the low-rank system.

One limitation of the graphon approach proposed is that each node in the graph is mapped to a single segment or point on the unit interval, meaning that each node can only be indexed by a single parameter. Embedded graph limit theory [2] gives an appropriate graph limit for graphs where each node has multiple parameters, or is otherwise distributed over a geometric space such as \mathcal{R}^2 or \mathcal{R}^3 . The incorporation of Q-noise into such graph limit systems may be shown to be straightforward.

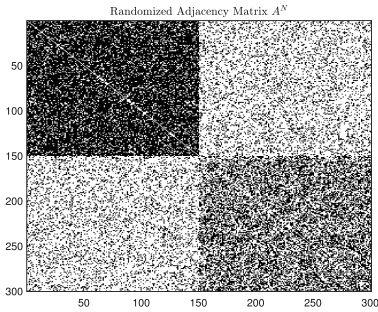


Fig. 14. The adjacency matrix of a 300 node stochastic block matrix. Similarly to the previous W-random graph, it is full rank.

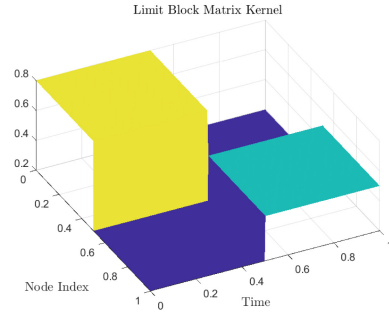


Fig. 15. The stochastic block matrix kernel A expressed as a piece-wise constant function on the unit square.

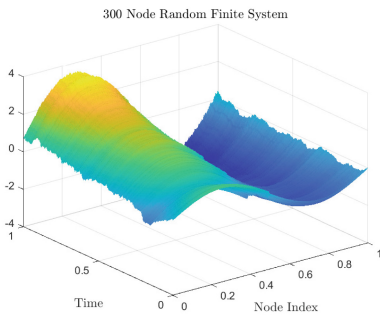


Fig. 16. The state trajectory associated with the randomized matrix A^N (Fig. 14).

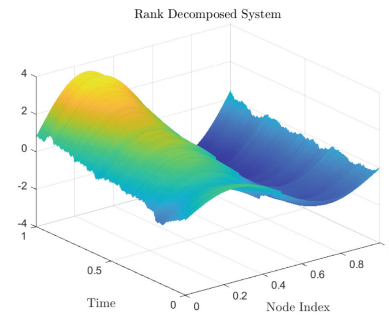


Fig. 17. The state trajectory of the graphon limit system, evaluated as two orthogonal components.

References

1. Aletti, G., Naldi, G.: Opinion dynamics on graphon: the piecewise constant case. *Appl. Math. Lett.* **133**, 108227 (2022). <https://linkinghub.elsevier.com/retrieve/pii/S0893965922001811>
2. Caines, P.E.: Embedded vertexon-graphons and embedded GMFG systems. In: 61st IEEE Conference on Decision and Control (2022)
3. Caines, P.E., Huang, M.: Graphon mean field games and the GMFG equations. In: 2018 IEEE Conference on Decision and Control (CDC), pp. 4129–4134 (2018). ISSN: 2576–2370
4. Caines, P.E., Huang, M.: Graphon mean field games and their equations. *SIAM J. Control Optim.* **59**(6), 4373–4399 (2021). <http://epubs.siam.org/doi/10.1137/20M136373X>

5. Calvó-Armengol, A., Jackson, M.O.: The effects of social networks on employment and inequality. *Am. Econ. Rev.* **94**(3), 426–454 (2004). <https://www.jstor.org/stable/3592937>, American Economic Association
6. Cauchemez, S., et al.: Role of social networks in shaping disease transmission during a community outbreak of 2009 H1N1 pandemic influenza. *Proc. Natl. Acad. Sci.* **108**(7), 2825–2830 (2011). <https://www.pnas.org/doi/abs/10.1073/pnas.1008895108>
7. Degroot, M.H.: Reaching a consensus. *J. Am. Stat. Assoc.* **69**(345), 118–121 (1974). <http://www.tandfonline.com/doi/abs/10.1080/01621459.1974.10480137>
8. Dunyak, A., Caines, P.E.: Large scale systems and SIR models: a featured graphon approach. In: 2021 60th IEEE Conference on Decision and Control (CDC), pp. 6928–6933 (2021). ISSN: 2576–2370
9. Dunyak, A., Caines, P.E.: Linear stochastic graphon systems with Q-space noise. In: 2022 IEEE 61st Conference on Decision and Control (CDC), pp. 3926–3932 (2022). ISSN: 2576–2370
10. Fabbri, G., Gozzi, F., Sweich, A.: *Stochastic Optimal Control in Infinite Dimension*. PTSM, vol. 82. Springer, Cham (2017). <https://doi.org/10.1007/978-3-319-53067-3>
11. Gao, S.: Fixed-point centrality for networks. In: 2022 IEEE 61st Conference on Decision and Control (CDC), pp. 1628–1635 (2022). ISSN: 2576–2370
12. Gao, S., Caines, P.E.: Spectral representations of graphons in very large network systems control. In: 2019 IEEE 58th Conference on Decision and Control (CDC), pp. 5068–5075 (2019). ISSN: 2576–2370
13. Gao, S., Caines, P.E.: Graphon control of large-scale networks of linear systems. *IEEE Trans. Autom. Control* **65**(10), 4090–4105 (2020)
14. Gao, S., Caines, P.E.: Subspace decomposition for graphon LQR: applications to VLSNs of harmonic oscillators. *IEEE Trans. Control Network Syst.* **8**(2), 576–586 (2021)
15. Gao, S., Caines, P.E., Huang, M.: LQG graphon mean field games: graphon invariant subspaces. In: 2021 60th IEEE Conference on Decision and Control (CDC), pp. 5253–5260 (2021). ISSN: 2576–2370
16. Gawarecki, L., Mandrekar, V.: *Stochastic Differential Equations in Infinite Dimensions. Probability and Its Applications*, Springer, Berlin Heidelberg, Berlin, Heidelberg (2011). <http://link.springer.com/10.1007/978-3-642-16194-0>
17. Gohberg, I., Goldberg, S.: *Basic Operator Theory*. Birkhäuser Boston, Boston, MA (1981). <http://link.springer.com/10.1007/978-1-4612-5985-5>
18. Holland, P.W., Laskey, K.B., Leinhardt, S.: Stochastic blockmodels: first steps. *Soc. Networks* **5**(2), 109–137 (1983). <https://www.sciencedirect.com/science/article/pii/0378873383900217>
19. Jackson, M.O.: *Social and economic networks*. In: *Social and Economic Networks*. Princeton University Press (2010). <https://www.degruyter.com/document/doi/10.1515/9781400833993/html>
20. Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., Boguñá, M.: Hyperbolic geometry of complex networks. *Phys. Rev. E* **82**(3), 036106 (2010). <https://link.aps.org/doi/10.1103/PhysRevE.82.036106>
21. Kukush, A.: *Gaussian Measures in Hilbert Space*. Wiley (2019). <http://onlinelibrary.wiley.com/doi/epub/10.1002/9781119476825>
22. Lovász, L.: *Large networks and graph limits* (2012). <https://bookstore.ams.org/coll-60/>

23. Medvedev, G.S., Simpson, G.: A numerical method for a nonlocal diffusion equation with additive noise. *Stochast. Partial. Diff. Equ. Anal. Comput.* (2022). <https://doi.org/10.1007/s40072-022-00262-w>
24. Tchuendom, R.F., Caines, P.E., Huang, M.: On the master equation for linear quadratic graphon mean field games. In: 2020 59th IEEE Conference on Decision and Control (CDC), pp. 1026–1031 (2020). ISSN: 2576–2370



Uniform Generation of Temporal Graphs with Given Degrees

Daniel Allendorf^(✉)

Goethe University Frankfurt, Frankfurt, Germany
dallendorf@ae.cs.uni-frankfurt.de

Abstract. Uniform sampling from the set $\mathcal{G}(\mathbf{d})$ of graphs with a given degree-sequence $\mathbf{d} = (d_1, \dots, d_n) \in \mathbb{N}^n$ is a classical problem in the study of random graphs. We consider an analogue for temporal graphs in which the edges are labeled with integer timestamps. The input to this generation problem is a tuple $\mathbf{D} = (\mathbf{d}, T) \in \mathbb{N}^n \times \mathbb{N}_{>0}$ and the task is to output a uniform random sample from the set $\mathcal{G}(\mathbf{D})$ of temporal graphs with degree-sequence \mathbf{d} and timestamps in the interval $[1, T]$. By allowing repeated edges with distinct timestamps, $\mathcal{G}(\mathbf{D})$ can be non-empty even if $\mathcal{G}(\mathbf{d})$ is, and as a consequence, existing algorithms are difficult to apply.

We describe an algorithm for this generation problem which runs in expected linear time $O(M)$ if $\Delta^{2+\epsilon} = O(M)$ for some constant $\epsilon > 0$ and $T - \Delta = \Omega(T)$ where $M = \sum_i d_i$ and $\Delta = \max_i d_i$. Our algorithm applies the switching method of McKay and Wormald [16] to temporal graphs: we first generate a random temporal *multigraph* and then remove self-loops and duplicated edges with switching operations which rewire the edges in a degree-preserving manner.

1 Introduction

A common problem in network science is the sampling of a graph matching a given degree-sequence. Formally, given a sequence of integers $\mathbf{d} = (d_1, \dots, d_n)$, we say that a graph $G = (V, E)$ with nodes $V = \{v_1, \dots, v_n\}$ matches \mathbf{d} if the number of incident edges at node v_i equals d_i for each $1 \leq i \leq n$. We then define $\mathcal{G}(\mathbf{d})$ as the set of all simple graphs (e.g. without loops or multi-edges) matching \mathbf{d} and ask for a uniform random sample $G \in \mathcal{G}(\mathbf{d})$. Such a sample is useful as it allows us to construct null models for testing the influence of the degrees on other graph properties of interest [4, 11]. In addition, this sampling problem is tightly related to the task of estimating $|\mathcal{G}(\mathbf{d})|$ [16, 17].

Temporal graphs are capable of modeling not only the topology but also the time structure of networks (see [14] or [6] for an overview). Possibly the most common type of temporal graph augments each edge of a classical graph with an integer timestamp. Here, we work by the following definition.

Definition 1 (Temporal Graph). A temporal (multi-)graph $G = (V, E)$ consists of a set of nodes $V = \{v_1, \dots, v_n\}$ and a (multi-)set of edges $E = \{e_1, \dots, e_m\}$ where each edge is a tuple $(\{u, v\}, t) \in \{\{u, v\} : u, v \in V\} \times \mathbb{N}_{>0}$.

Long version with proof details available at <https://arxiv.org/abs/2304.09654>.

In terms of semantics, the presence of an edge $(\{u, v\}, t)$ indicates that the nodes u and v are connected at time t . For the purpose of modeling networks, it additionally makes sense to restrict ourselves to *simple* temporal graphs which exclude certain types of edges. To this end, we call a temporal graph $G = (V, E)$ *simple* if the edge set E contains no loops and no edges between the same nodes with the same timestamp, i.e. iff $u \neq v$ for all $(\{u, v\}, t) \in E$ and E is a set.

Given its prominence in classical graph theory, it is reasonable to assume that parameterizing temporal graphs by the degrees can yield a similarly useful model. For instance, the distribution of active times of nodes and edges in real-world temporal graphs has been observed to follow a power-law [2, 13, 14], which can be reproduced by sampling a uniform random temporal graph with power-law degrees. Still, to the best of our knowledge, the problem of generating such graphs has not been considered so far. In this paper, we study algorithmic techniques of sampling such graphs uniformly at random, and in particular, focus on the task of providing an exact uniform sample.

1.1 Related Work

A simple way to obtain a uniform random sample $G \in \mathcal{G}(\mathbf{d})$ is to use the *configuration model* of Bender and Canfield [3] or Bollobás [5] to sample random *multigraphs* with sequence \mathbf{d} until a simple graph is found. Unfortunately, this simple rejection scheme is not efficient as its run time is exponential in the largest degree $\Delta = \max_i d_i$ [18]. More efficient algorithms have been obtained via the *switching* method of McKay and Wormald [16]. The approach is to again start from a random multigraph but instead of rejecting non-simple graphs, loops and multi-edges are removed with switchings which rewire the edges while preserving the degrees. In addition to an algorithm with expected runtime $O(m)$ for generating graphs with m edges and bounded degrees $\Delta^4 = O(m)$ [1, 16], efficient algorithms have been given for d -regular graphs in expected time $O(nd + d^4)$ if $d = o(\sqrt{n})$ [1, 9], and graphs with power-law degrees in expected time $O(n)$ if the exponent satisfies $\gamma > (21 + \sqrt{61})/10$ [1, 10]. Alternatively, there exist efficient solutions to various relaxations of the problem. For instance, we may allow the graph to match the sequence \mathbf{d} only in expectation [7], or use a Markov chain to approximate the uniform distribution [8, 15]. See also [12] for a survey.

1.2 Our Contribution

We give results on sampling temporal graphs with given degrees and lifetime. Formally, given a tuple $\mathbf{D} = (\mathbf{d}, T)$, we say that a temporal graph G with nodes $V = \{v_1, \dots, v_n\}$ matches \mathbf{D} if the sum of the numbers of incident edges at node v_i over all T timestamps equals d_i for each $1 \leq i \leq n$. If at least one simple graph matches \mathbf{D} , we call \mathbf{D} *realizable*. The temporal graph generation problem now asks to output a sample $G \in \mathcal{G}(\mathbf{D})$ uniformly at random from the set $\mathcal{G}(\mathbf{D})$ of matching simple temporal graphs. Note that by allowing repeated connections, a given tuple $\mathbf{D} = (\mathbf{d}, T)$ can be realizable as a simple temporal graph even if \mathbf{d} is not realizable as a classical simple graph. More severely, consider the sequence

$\mathbf{d}_t, 1 \leq t \leq T$ of degree-sequences induced by the individual timestamps, then there exist sequences which satisfy $\sum_t \mathbf{d}_t = \mathbf{d}$ but are not realizable as a sequence of simple graphs due to loops or multi-edges which cannot be rewired with switchings which preserve $\mathbf{d}_t, 1 \leq t \leq T$. In other words, even distributing the degrees among timestamps is not trivial. Existing algorithms are therefore difficult to apply to sampling temporal graphs. Instead, switchings are required which operate on the temporal graph as a whole and re-assign timestamps where necessary, and this is the key feature of the algorithm which we describe here. This algorithm, called T-GEN, generates simple temporal graphs with bounded degrees. Our main result is as follows (proof given in the long version).

Theorem 1. *Given a realizable tuple $\mathbf{D} = (\mathbf{d}, T)$ which satisfies $\Delta^{2+\epsilon} = O(M)$ for a constant $\epsilon > 0$ and $T - \Delta = \Omega(T)$, T-GEN outputs a uniform random sample $G \in \mathcal{G}(\mathbf{D})$ in expected time $O(M)$.*

1.3 Overview of Techniques

The general idea is to apply the switching method of [16] to temporal graphs. To this end, we define a *temporal configuration model* (see Sect. 2) which samples a random temporal multigraph with the property that the probability of a given graph only depends on the contained loops and temporal multi-edges. This allows us to preserve uniformity when applying switchings and ensures the uniformity of the final output. Still, a challenge remains in that the sequences $\mathbf{d}_t, 1 \leq t \leq T$ implied by the random temporal multigraph may not be realizable. This necessitates switchings which rewire edges across different time slices of the graph and re-assign timestamps (see Definition 2 for an example). As a consequence, the number of timestamps we can assign to an edge without creating a temporal multi-edge affects the distribution of the graphs, and to preserve uniformity, it becomes necessary to account for the available timestamps. To discuss this matter, we briefly describe the technique used in [16] to correct the distribution.

Generally speaking, when analyzing a switching operation θ we fix subsets $\mathcal{S}, \mathcal{S}' \subseteq \mathcal{M}(\mathbf{d})$ of the set $\mathcal{M}(\mathbf{d})$ of multigraphs matching the sequence \mathbf{d} . Considering the edges rewired by θ then associates each graph $G \in \mathcal{S}$ with a subset $\mathcal{F}(G) \subseteq \mathcal{S}'$ of graphs in \mathcal{S}' which can be produced by a type θ switching on G , and each graph $G' \in \mathcal{S}'$ with a subset $\mathcal{B}(G') \subseteq \mathcal{S}$ of graphs on which a type θ switching can produce G' . If we now start from a uniform random graph $G \in \mathcal{S}$, and perform a uniform random θ switching on G , the probability of reaching a given graph $G' \in \mathcal{S}'$ is

$$\sum_{G \in \mathcal{B}(G')} \frac{1}{|\mathcal{S}| |\mathcal{F}(G)|}$$

which depends on G and G' . To correct this, *rejection* steps are used which restart the algorithm with a certain probability. Before performing the switching, we *f-reject* (forward reject) with probability $1 - |\mathcal{F}(G)|/\bar{f}(\mathcal{S})$ where $\bar{f}(\mathcal{S})$ is an upper bound on $|\mathcal{F}(G)|$ over all graphs $G \in \mathcal{S}$, and after performing the switching, we *b-reject* (backward reject) with probability $1 - \underline{b}(\mathcal{S}')/|\mathcal{B}(G')|$ where $\underline{b}(\mathcal{S}')$ is

a lower bound on $|\mathcal{B}(G')|$ over all graphs $G' \in \mathcal{S}'$. The probability of producing G' is now

$$\sum_{G \in \mathcal{B}(G')} \frac{1}{|\mathcal{S}'| |\mathcal{F}(G)|} \frac{|\mathcal{F}(G)|}{\bar{f}(\mathcal{S})} \frac{b(\mathcal{S}')}{|\mathcal{B}(G')|} = \frac{|\mathcal{B}(G')|}{|\mathcal{S}'| \bar{f}(\mathcal{S})} \frac{b(\mathcal{S}')}{|\mathcal{B}(G')|} = \frac{b(\mathcal{S}')}{|\mathcal{S}'| \bar{f}(\mathcal{S})},$$

implying that G' has the uniform distribution if G does. Still, this method of correcting the distribution is efficient only if $|\mathcal{F}(G)|$ and $|\mathcal{B}(G')|$ do not deviate too much from $\bar{f}(\mathcal{S})$ and $b(\mathcal{S}')$. To avoid a high probability of restarting where this does not hold, Gao and Wormald [9] first used additional switchings which equalize the probabilities by mapping high probability graphs to low probability graphs. This is done via a Markov chain which either chooses a main switching to remove a non-simple edge or an additional switching to equalize the probabilities.

T-GEN similarly uses additional switchings but without the use of a Markov chain. Instead, we always perform a main switching first and then an additional switching which targets specific edges involved in the main switching. Concretely, the issue is that the typical number of available timestamps for an edge is $\Omega(T)$, whereas the lower bound is $T - (\Delta - 1)$ due to graphs in which the edge has multiplicity Δ . Still, the conditions imposed in Theorem 1 imply that the highest multiplicity of any edge in any graph visited by the algorithm is bound by a constant μ with high probability. Now, after performing a main kind of switching, we partition the set \mathcal{S}' which contains the obtained graph into subsets $\mathcal{S}'_{\mathbf{m} < \mu}$ and $\mathcal{S}' \setminus \mathcal{S}'_{\mathbf{m} < \mu}$ by the multiplicities \mathbf{m} of specific edges involved in the switching. We then equalize the probabilities of producing the graphs in $\mathcal{S}'_{\mathbf{m} < \mu}$ via switchings which involve the specific edges with the standard rejection step (which is efficient by $\mu = O(1)$). To equalize the probabilities between the graphs in $\mathcal{S}'_{\mathbf{m} < \mu}$ and $\mathcal{S}' \setminus \mathcal{S}'_{\mathbf{m} < \mu}$, we define auxiliary switching operations which map the graphs in $\mathcal{S}'_{\mathbf{m} < \mu}$ to graphs in $\mathcal{S}' \setminus \mathcal{S}'_{\mathbf{m} < \mu}$ and an identity switching which maps any graph in $\mathcal{S}'_{\mathbf{m} < \mu}$ to itself, and specify a probability distribution over these two kinds of switchings which ensures that all graphs in \mathcal{S}' are produced with the same probability via switchings which involve the specific edges.

2 Temporal Configuration Model

The temporal configuration model samples a random temporal multigraph matching a given tuple $\mathbf{D} = (\mathbf{d}, T) \in \mathbb{N}^n \times \mathbb{N}_{>0}$. It can be implemented as follows. First, for each node index $i \in \{1, \dots, n\}$, put d_i marbles labeled i into an urn. Then, starting from the empty graph $G = (V, \emptyset)$ on the node set $V = \{v_1, \dots, v_n\}$, add edges by iteratively performing the following steps until the urn is empty:

1. Draw two marbles from the urn uniformly at random (without replacement), and let i, j denote the labels of those marbles.
2. Draw a timestamp t uniformly at random from the set of timestamps $[1, T]$.
3. Add the temporal edge $(\{v_i, v_j\}, t)$ to the graph G .

In the following, we analyze the output distribution of this random model. To this end, we first give some definitions to characterize the edges in a temporal multigraph. Given two nodes $v_i, v_j \in V$ and a timestamp $t \in [1, T]$, we define $w_{i,j,t}$ as the number of edges between v_i and v_j with timestamp t in the graph, and call $w_{i,j,t}$ the *temporal multiplicity* of the edge $(\{v_i, v_j\}, t)$. Then, if $w_{i,j,t} \geq 2$, we say that the edge is contained in a *temporal multi-edge*, and in the special cases $w_{i,j,t} = 2$ and $w_{i,j,t} = 3$, refer to the multi-edge as a *double-edge* and *triple-edge*, respectively. In addition, we define $m_{i,j} = \sum_t w_{i,j,t}$ as the total number of edges between v_i and v_j over all timestamps and call $m_{i,j}$ the *multiplicity* of $\{v_i, v_j\}$. Finally, we call an edge $(\{v_i\}, t)$ which connects a node v_i to itself a *loop* at v_i , and in the cases where $w_{i,t} = 1$ and $w_{i,t} = 2$, refer to the edge as a *temporal single-loop* and *temporal double-loop*, respectively.

Now, for a given temporal multigraph G , let $\mathbf{W}(G)$ denote the $n \times n \times T$ tensor such that the entries $\mathbf{W}_{i,j,t}(G)$ where $i \neq j$ contain $w_{i,j,t}$ if $w_{i,j,t} \geq 2$ and 0 if otherwise, and the entries $\mathbf{W}_{i,i,t}(G)$ contain $w_{i,t}$. In addition, let $\mathcal{M}(\mathbf{D})$ denote the set of temporal multigraphs matching a given tuple \mathbf{D} , and for a given tensor \mathbf{W} , let $\mathcal{S}(\mathbf{W})$ denote the subset of temporal multigraphs $G \in \mathcal{M}(\mathbf{D})$ such that $\mathbf{W}(G) = \mathbf{W}$. Then, the following holds (proof in the long version).

Theorem 2. *Let G be a temporal multigraph output by the temporal configuration model on an input tuple \mathbf{D} . Then, G is uniformly distributed in the set $\mathcal{S}(\mathbf{W}(G)) \subseteq \mathcal{M}(\mathbf{D})$.*

3 Algorithm T-Gen

T-GEN takes a realizable tuple \mathbf{D} as input and outputs a uniform random sample $G \in \mathcal{G}(\mathbf{D})$ from the set of matching simple temporal graphs. The algorithm starts by sampling a temporal multigraph $G \in \mathcal{M}(\mathbf{D})$ via the temporal configuration model (see Sect. 2). It then checks if G satisfies initial conditions on the numbers and multiplicities of non-simple edges (see Subsect. 3.1). In particular, G is not allowed to contain temporal triple-edges, or temporal double-loops (or higher multiplicities). If G satisfies these conditions, then the algorithm proceeds to removing all temporal single-loops and temporal double-edges during two stages. Stage 1 (Subsect. 3.2) removes all temporal single-loops. For this purpose three kinds of switching operations are used. The main kind of switching removes a temporal single-loop at a specified node and with a specified timestamp. After performing this kind of switching we always perform one of two auxiliary switchings. The purpose of these switchings is to equalize the probabilities between graphs which contain edges of high multiplicity and graphs which do not. Stage 2 (Subsect. 3.3) removes all temporal double-edges. Doing this efficiently requires five kinds of switchings, two of which remove temporal double-edges and three of which are auxiliary switchings. Once all non-simple edges have been removed, the resulting graph is output.

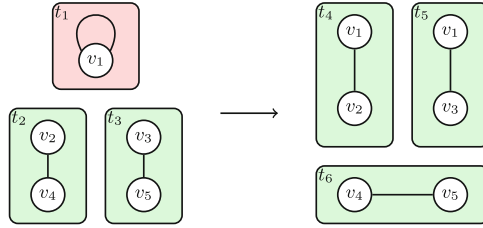


Fig. 1. The TL switching removes a temporal single-loop with timestamp t_1 at a node v_1 . Red and green shades indicate non-simple and simple edges.

3.1 Initial Conditions

The initial conditions for the random multigraph G are as follows. Define

$$B_L = \frac{M_2}{M}, \quad B_D = \frac{M_2^2}{M^2 T}, \quad \text{where} \quad M = \sum_i d_i, \quad M_2 = \sum_i d_i(d_i - 1),$$

let $L = \sum_{i,t} \mathbf{W}_{i,i,t}(G)$ and $D = \sum_{i \neq j,t} \mathbf{W}_{i,j,t}(G)$ denote the sums of the multiplicities of loops and temporal multi-edges of G , respectively, and choose three constants

$$\lambda \geq 1 + 1/\epsilon, \quad \kappa \geq 1 + 1/\epsilon, \quad \mu \geq 3 + 2/\epsilon$$

where $\epsilon > 0$ is a constant such that $\Delta^{2+\epsilon} = O(M)$ (set $\lambda = \kappa = \mu = \Delta$ if no such constant exists). Then, G satisfies the initial conditions if $L \leq B_L$, $D/2 \leq B_D$, there are no loops of temporal multiplicity $w \geq 2$ or temporal multi-edges of temporal multiplicity $w \geq 3$, and no node is incident with more than λ temporal single-loops or κ temporal double-edges.

If the graph G does not satisfy the conditions, then T-GEN restarts. Otherwise, the algorithm enters Stage 1 to remove the temporal single-loops in G .

3.2 Stage 1: Removal of Temporal Single-Loops

Stage 1 removes all temporal single-loops in the graph. Doing this efficiently requires three kinds of switchings which we denote as TL, $A_{m,n}$ and I. The switching of the main kind written as TL removes a temporal single-loop at a specified node and with a specified timestamp. After performing this kind of switching, an $A_{m,n}$ auxiliary switching is performed with a certain probability. This switching adds up to two edges with multiplicities $\max\{m, n\} \geq \mu$ to the graph to equalize the probability of producing graphs with or without these kinds of edges. In addition, we define the identity switching I which maps each graph to itself. Formal definitions of the TL and $A_{m,n}$ switchings are as follows.

Definition 2 (TL switching at v_1, t_1). For a graph G such that $(\{v_1\}, t_1)$ is a temporal single-loop, let $(\{v_2, v_4\}, t_2)$, $(\{v_3, v_5\}, t_3)$ be edges and $t_4, t_5, t_6 \in [1, T]$ timestamps such that

- none of the edges $(\{v_2, v_4\}, t_2)$, $(\{v_3, v_5\}, t_3)$ is a loop or in a temporal double-edge,
- the nodes v_2, v_3, v_4, v_5 are distinct from v_1 , and v_4 is distinct from v_5 , and
- none of the edges $(\{v_1, v_2\}, t_4)$, $(\{v_1, v_3\}, t_5)$, $(\{v_4, v_5\}, t_6)$ exist.

Then, a TL switching replaces the edges $(\{v_1\}, t_1)$, $(\{v_2, v_4\}, t_2)$, $(\{v_3, v_5\}, t_3)$ with $(\{v_1, v_2\}, t_4)$, $(\{v_1, v_3\}, t_5)$, $(\{v_4, v_5\}, t_6)$ (see Fig. 1).

Definition 3 ($A_{m,n}$ switching at v_1, v_2, v_3, v_4, v_5). For a graph G such that $\{v_2, v_4\}$, $\{v_3, v_5\}$ are non-edges, let $(\{v_2, v_{2i+4}\}, t_i)$, $(\{v_4, v_{2i+5}\}, t_{m+i})$, $1 \leq i \leq m$ be incident edges at v_2, v_4 , $(\{v_3, v_{2m+2i+4}\}, t_{2m+i})$, $(\{v_5, v_{2m+2i+5}\}, t_{2m+n+i})$, $1 \leq i \leq n$ incident edges at v_3, v_5 , and $t_{2m+2n+1}, \dots, t_{4m+4n} \in [1, T]$ timestamps such that

- none of the edges is a loop or in a temporal double-edge,
- the nodes $v_1, \dots, v_{2m+2n+5}$ are all distinct, and
- none of the edges $(\{v_2, v_4\}, t_{2m+2n+i})$, $(\{v_{2i+4}, v_{2i+5}\}, t_{3m+2n+i})$, $1 \leq i \leq m$ and $(\{v_3, v_5\}, t_{4m+2n+i})$, $(\{v_{2m+2i+4}, v_{2m+2i+5}\}, t_{4m+3n+i})$, $1 \leq i \leq n$ exist.

Then, an $A_{m,n}$ switching replaces the edges $(\{v_2, v_{2i+4}\}, t_i)$, $(\{v_4, v_{2i+5}\}, t_{m+i})$, $1 \leq i \leq m$, $(\{v_3, v_{2m+2i+4}\}, t_{2m+i})$, $(\{v_3, v_{2m+2i+5}\}, t_{2m+n+i})$, $1 \leq i \leq n$ with $(\{v_2, v_4\}, t_{2m+2n+i})$, $(\{v_{2i+4}, v_{2i+5}\}, t_{3m+2n+i})$, $1 \leq i \leq m$, $(\{v_3, v_5\}, t_{4m+2n+i})$, $(\{v_{2m+2i+4}, v_{2m+2i+5}\}, t_{4m+3n+i})$, $1 \leq i \leq n$.

In other words, the TL switching chooses two edges and then rewires the specified loop and the two edges such that exactly the specified loop is removed and no other non-simple edges are created or removed. Likewise, the $A_{m,n}$ switching chooses m incident edges at two nodes v_2, v_4 each and n incident edges at two nodes v_3, v_5 each and then rewires the edges such that exactly m simple edges between the nodes v_2, v_4 and exactly n simple edges between the nodes v_3, v_5 are created and no non-simple edges are created or removed.

After each TL switching, we perform an $A_{m,n}$ auxiliary switching or the identity switching. To decide which switching to perform, we define a probability distribution over the switchings which ensures uniformity. In total, the set of $A_{m,n}$ auxiliary switchings is $\Theta_A = \bigcup_{0 \leq m, n < \Delta, \mu \leq \max\{m, n\}} \{A_{m,n}\}$. A switching is then sampled from the distribution $(\Theta_A \cup \{I\}, P_A)$ where

$$P_A(A_{m,n}) = P_A(I) \frac{\bar{f}_{A_{m,n}}(\mathbf{W}')}{\underline{b}_{A_{m,n}}(\mathbf{W}')}, \quad P_A(I) = 1 - \sum_{\theta \in \Theta_A} P_A(\theta)$$

for quantities $\bar{f}_{A_{m,n}}(\mathbf{W}')$ and $\underline{b}_{A_{m,n}}(\mathbf{W}')$ given further below.

On a high level, Stage 1 runs in a loop until a rejection occurs or all temporal single-loops have been removed from G . To this end, let π denote a permutation of the entries in $\mathbf{W}(G)$ such that $\mathbf{W}_{i,i,t} = 1$. Then, Stage 1 iterates through the temporal single-loops in the order given by π and performs the following steps for each temporal single-loop.

1. Let G denote the current graph, $\mathbf{W} = \mathbf{W}(G)$ and $(\{v_1\}, t_1)$ the loop.
2. Pick a uniform random TL switching S which removes $(\{v_1\}, t_1)$ from G .
3. Restart (**f-reject**) with probability $1 - \frac{f_{\text{TL}}(G)}{\bar{f}_{\text{TL}}(\mathbf{W})}$.
4. Rewire the edges according to S , let G' denote the result and $\mathbf{W}' = \mathbf{W}(G')$.
5. Let $(\{v_2, v_4\}, t_2)$, $(\{v_3, v_5\}, t_3)$ denote the edges removed by S .
6. Restart if $m_{2,4} \geq \mu$ or $m_{3,5} \geq \mu$.
7. Restart (**b-reject**) with probability $1 - \frac{b_{\text{TL}}(\mathbf{W}'; 2)}{b_{\text{TL}}(G', v_1 v_2 v_3 v_4 v_5; 2)}$.
8. Choose a switching type $\theta \sim (\Theta_A \cup \{1\}, P_A)$.
9. If $\theta = A_{m,n}$ for some $A_{m,n} \in \Theta_A$:
 - (a) Restart if $m_{2,4} \geq 1$ or $m_{3,5} \geq 1$.
 - (b) Pick a uniform random $A_{m,n}$ switching S' on G' .
 - (c) Restart (**f-reject**) with probability $1 - \frac{f_{A_{m,n}}(G')}{\bar{f}_{A_{m,n}}(\mathbf{W}')}$.
 - (d) Rewire the edges according to S' and let G'' denote the result.
 - (e) Restart (**b-reject**) with probability $1 - \frac{b_{A_{m,n}}(\mathbf{W}')}{b_{A_{m,n}}(G'', v_1 v_2 v_3 v_4 v_5)}$.
 - (f) Set $G' \leftarrow G''$.
10. Restart (**b-reject**) with probability $1 - \frac{b_{\text{TL}}(\mathbf{W}'; 0) b_{\text{TL}}(\mathbf{W}'; 1)}{b_{\text{TL}}(G', v_1; 0) b_{\text{TL}}(G', v_1 v_2 v_3; 1)}$.
11. Set $G \leftarrow G'$.

To fully specify Stage 1, it remains to define the quantities for the f- and b-rejection steps. For the f-rejection in step 3, define $f_{\text{TL}}(G)$ as the number of TL switchings which can be performed on G . The corresponding upper bound is

$$\bar{f}_{\text{TL}}(\mathbf{W}) = M^2 T^3.$$

For the b-rejections in steps 7 and 10, define $b_{\text{TL}}(G', v_1 v_2 v_3 v_4 v_5; 2)$ as the number of timestamps $t_2, t_3 \in [1, T]$ such that the edges $(\{v_2, v_4\}, t_2)$, $(\{v_3, v_5\}, t_3)$ do not exist in G' , $b_{\text{TL}}(G', v_1 v_2 v_3; 1)$ as the number of simple temporal edges $(\{v_4, v_5\}, t_6)$ such that v_4, v_5 are distinct from v_1, v_2, v_3 , and $b_{\text{TL}}(G', v_1; 0)$ as the number of distinct simple temporal edges $(\{v_1, v_2\}, t_4)$, $(\{v_1, v_3\}, t_5)$ incident at v_1 . The lower bounds on these quantities are

$$\begin{aligned} \underline{b}_{\text{TL}}(\mathbf{W}'; 2) &= (T - (\mu - 1))^2, & \underline{b}_{\text{TL}}(\mathbf{W}'; 1) &= M - 2B_L - 4B_D - 4\Delta, \\ \underline{b}_{\text{TL}}(\mathbf{W}'; 0) &= k_1(k_1 - 1) \end{aligned}$$

where $k_i = d_i - \sum_t (2\mathbf{W}'_{i,i,t} + \sum_{1 \leq j \leq n: j \neq i} \mathbf{W}'_{i,j,t})$ denotes the number of simple edges at v_i . For the f-rejection in step 9c, define $f_{A_{m,n}}(G')$ as the number of $A_{m,n}$ switchings which can be performed on the graph G' . The upper bound is

$$\bar{f}_{A_{m,n}}(\mathbf{W}') = (\Delta T)^{2(m+n)}.$$

For the b-rejection in step 9e, define $b_{A_{m,n}}(G'', v_1 v_2 v_3 v_4 v_5)$ as the number of $A_{m,n}$ switchings which can produce the graph G'' . The corresponding lower bound is

$$\underline{b}_{A_{m,n}}(\mathbf{W}') = ((M - 2B_L - 4B_D - 4(m+n+3)\Delta)(T - (\Delta - 1))^2)^{m+n}.$$

Stage 1 ends if all temporal single-loops have been removed. The algorithm then moves on to Stage 2 to remove the remaining temporal double-edges.

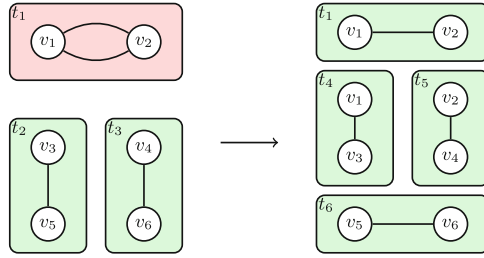


Fig. 2. The TD_1 switching removes a temporal double-edge with timestamp t_1 between two nodes v_1, v_2 while leaving a single-edge between the nodes.

3.3 Stage 2: Removal of Temporal Double-Edges

Stage 2 uses five kinds of switchings: TD_1 , TD_0 , $\mathcal{B}_{m,n}$, $\mathcal{C}_{m,n,o,p}$ and \mathcal{I} . The two main switchings TD_1 and TD_0 remove a temporal double-edge between two specified nodes with a specified timestamp. However, the TD_1 switching removes one occurrence of the edge while the TD_0 switching removes both occurrences. This is done to equalize the probability between graphs in which the edge is a non-edge, or single edge. After performing a TD_1 switching, a $\mathcal{B}_{m,n}$ auxiliary switching may be performed, and after performing a TD_0 switching, a $\mathcal{C}_{m,n,o,p}$ switching may be performed. These switchings add edges with multiplicity $\max\{m, n\} \geq \mu$ or $\max\{m, n, o, p\} \geq \mu$ to the graph to equalize the probabilities between graphs with or without these edges. The TD_1 switching is defined as follows.

Definition 4 (TD_1 switching at $(\{v_1, v_2\}, t_1)$). For a graph G such that $(\{v_1, v_2\}, t_1)$ is contained in a temporal double-edge, let $(\{v_3, v_5\}, t_2)$, $(\{v_4, v_6\}, t_3)$ be edges and $t_4, t_5, t_6 \in [1, T]$ timestamps such that

- none of the edges $(\{v_3, v_5\}, t_2)$, $(\{v_4, v_6\}, t_3)$ is in a temporal double-edge,
- v_3, v_4, v_5, v_6 are distinct from v_1 and v_2 , and v_5 is distinct from v_6 , and
- none of the edges $(\{v_1, v_3\}, t_4)$, $(\{v_2, v_4\}, t_5)$, $(\{v_5, v_6\}, t_6)$ exist.

Then, a TD_1 switching replaces the edges $(\{v_1, v_2\}, t_1)$, $(\{v_3, v_5\}, t_2)$, $(\{v_4, v_6\}, t_3)$ with $(\{v_1, v_3\}, t_4)$, $(\{v_2, v_4\}, t_5)$, $(\{v_5, v_6\}, t_6)$ (see Fig. 2).

The TD_0 switching can be defined analogously by using four edges instead of two to remove both edges contained in the temporal double-edge. To choose a TD_1 or TD_0 switching when removing a temporal double-edge, specify the probability distribution $(\{\text{TD}_1, \text{TD}_0\}, P)$ where

$$p(\text{TD}_1) = p(\text{TD}_0) \frac{p_{\mathcal{C}}(\mathcal{I}) \overline{f}_{\text{TD}_1}(\mathbf{W}) \underline{b}_{\text{TD}_0}(\mathbf{W}')}{p_{\mathcal{B}}(\mathcal{I}) \overline{f}_{\text{TD}_0}(\mathbf{W}) \underline{b}_{\text{TD}_1}(\mathbf{W}')}, \quad p(\text{TD}_0) = 1 - p(\text{TD}_1)$$

for quantities defined further below.

Continuing with the auxiliary switchings, we now define the $\mathcal{B}_{m,n}$ switching. The $\mathcal{C}_{m,n,o,p}$ switching can be defined analogously by extending the edges created from the two edges removed by the TD_1 switching to the four edges removed by the TD_0 switching.

Definition 5 ($\mathbf{B}_{m,n}$ switching at $v_1, v_2, v_3, v_4, v_5, v_6$) For a graph G such that $\{v_3, v_5\}, \{v_4, v_6\}$ are non-edges, let $(\{v_3, v_{2i+5}\}, t_i), (\{v_5, v_{2i+6}\}, t_{m+i}), 1 \leq i \leq m$ be incident edges at $v_3, v_5, (\{v_4, v_{2m+2i+5}\}, t_{2m+i}), (\{v_6, v_{2m+2i+6}\}, t_{2m+n+i}), 1 \leq i \leq n$ incident edges at v_4, v_6 , and $t_{2m+2n+1}, \dots, t_{4m+4n} \in [1, T]$ timestamps such that

- none of the edges is in a temporal double-edge,
- the nodes $v_1, \dots, v_{2m+2n+6}$ are all distinct, and
- none of the edges $(\{v_3, v_5\}, t_{2m+2n+i}), (\{v_{2i+5}, v_{2i+6}\}, t_{3m+2n+i}), 1 \leq i \leq m$ and $(\{v_4, v_6\}, t_{4m+2n+i}), (\{v_{2m+2i+5}, v_{2m+2i+6}\}, t_{4m+3n+i}), 1 \leq i \leq n$ exist.

Then, a $\mathbf{B}_{m,n}$ switching replaces the edges $(\{v_3, v_{2i+5}\}, t_i), (\{v_5, v_{2i+6}\}, t_{m+i}), 1 \leq i \leq m, (\{v_4, v_{2m+2i+5}\}, t_{2m+i}), (\{v_6, v_{2m+2i+6}\}, t_{2m+n+i}), 1 \leq i \leq n$ with $(\{v_3, v_5\}, t_{2m+2n+i}), (\{v_{2i+5}, v_{2i+6}\}, t_{3m+2n+i}), 1 \leq i \leq m, (\{v_4, v_6\}, t_{4m+2n+i}), (\{v_{2m+2i+5}, v_{2m+2i+6}\}, t_{4m+3n+i}), 1 \leq i \leq n$.

The sets of $\mathbf{B}_{m,n}$ and $\mathbf{C}_{m,n,o,p}$ switchings are $\Theta_{\mathbf{B}} = \bigcup_{0 \leq m, n < \Delta, \mu \leq \max\{m, n\}} \{\mathbf{B}_{m,n}\}$ and $\Theta_{\mathbf{C}} = \bigcup_{0 \leq m, n, o, p < \Delta, \mu \leq \max\{m, n, o, p\}} \{\mathbf{C}_{m,n,o,p}\}$. The associated type distributions are $(\Theta_{\mathbf{B}} \cup \{\mathbf{I}\}, P_{\mathbf{B}})$ and $(\Theta_{\mathbf{C}} \cup \{\mathbf{I}\}, P_{\mathbf{C}})$ where

$$P_{\mathbf{B}}(\mathbf{B}_{m,n}) = P_{\mathbf{B}}(\mathbf{I}) \frac{\bar{f}_{\mathbf{B}_{m,n}}(\mathbf{W}')}{\underline{b}_{\mathbf{B}_{m,n}}(\mathbf{W}')}, \quad P_{\mathbf{B}}(\mathbf{I}) = 1 - \sum_{\theta \in \Theta_{\mathbf{B}}} P_{\mathbf{B}}(\theta),$$

$$P_{\mathbf{C}}(\mathbf{C}_{m,n,o,p}) = P_{\mathbf{C}}(\mathbf{I}) \frac{\bar{f}_{\mathbf{C}_{m,n,o,p}}(\mathbf{W}')}{\underline{b}_{\mathbf{C}_{m,n,o,p}}(\mathbf{W}')}, \quad P_{\mathbf{C}}(\mathbf{I}) = 1 - \sum_{\theta \in \Theta_{\mathbf{C}}} P_{\mathbf{C}}(\theta)$$

for quantities defined further below.

The main loop of Stage 2 is as follows. Let π denote a permutation of the entries in $\mathbf{W}(G)$ such that $\mathbf{W}_{i,j,t} = 2$ and $i \neq j$. Then, Stage 2 iterates through the temporal double-edges in the order of π and performs the following steps.

1. Let G be the graph, $\mathbf{W} = \mathbf{W}(G)$ and $(\{v_1, v_2\}, t_1)$ the temporal double-edge.
2. Choose a switching type $\theta \sim (\{\text{TD}_0, \text{TD}_1\}, P)$.
3. Pick a uniform random θ switching S which removes $(\{v_1, v_2\}, t_1)$ from G .
4. Restart (**f-reject**) with probability $1 - \frac{f_{\theta}(G)}{f_{\theta}(\mathbf{W})}$.
5. Rewire the edges according to S , let G' denote the result and $\mathbf{W}' = \mathbf{W}(G')$.
6. If $\theta = \text{TD}_1$:
 - (a) Let $(\{v_3, v_5\}, t_2), (\{v_4, v_6\}, t_3)$ denote the edges removed by S .
 - (b) Restart if $m_{3,5} \geq \mu$ or $m_{4,6} \geq \mu$.
 - (c) Restart (**b-reject**) with probability $1 - \frac{b_{\text{TD}_0}(\mathbf{W}';2)}{b_{\text{TD}_0}(G', v_1 \dots v_6; 2)}$.
 - (d) Choose a switching type $\theta_{\mathbf{B}} \sim (\Theta_{\mathbf{B}} \cup \{\mathbf{I}\}, P_{\mathbf{B}})$.
 - (e) If $\theta = \mathbf{B}_{m,n}$ for some $\mathbf{B}_{m,n} \in \Theta_{\mathbf{B}}$:
 - i. Restart if $m_{3,5} \geq 1$ or $m_{4,6} \geq 1$.
 - ii. Pick a uniform random $\mathbf{B}_{m,n}$ switching S' on G' .
 - iii. Restart (**f-reject**) with probability $1 - \frac{f_{\mathbf{B}_{m,n}}(G')}{\bar{f}_{\mathbf{B}_{m,n}}(\mathbf{W}')}.$

- iv. Rewire the edges according to S' and let G'' denote the result.
 - v. Restart (**b-reject**) with probability $1 - \frac{\underline{b}_{B_{m,n}}(\mathbf{W}')}{\underline{b}_{B_{m,n}}(G'', v_1 \dots v_6)}$.
 - vi. Set $G' \leftarrow G''$.
 - (f) Restart (**b-reject**) with probability $1 - \frac{\underline{b}_{TD_1}(\mathbf{W}'; 0) \underline{b}_{TD_1}(\mathbf{W}'; 1)}{\underline{b}_{TD_1}(G', v_1 v_2; 0) \underline{b}_{TD_1}(G', v_1 v_2 v_3 v_4; 1)}$.
7. Else if $\theta = TD_0$:
- (a) Let $(\{v_3, v_7\}, t_2)$, $(\{v_4, v_8\}, t_3)$, $(\{v_5, v_9\}, t_4)$, $(\{v_6, v_{10}\}, t_5)$ denote the edges removed by S .
 - (b) Restart if $m_{3,7} \geq \mu$, $m_{4,8} \geq \mu$, $m_{5,9} \geq \mu$ or $m_{6,10} \geq \mu$.
 - (c) Restart (**b-reject**) with probability $1 - \frac{\underline{b}_{TD_0}(\mathbf{W}'; 2)}{\underline{b}_{TD_0}(G', v_1 \dots v_{10}; 2)}$.
 - (d) Choose a switching type $\theta_C \sim (\Theta_C \cup \{1\}, P_C)$.
 - (e) If $\theta = C_{m,n,o,p}$ for some $C_{m,n,o,p} \in \Theta_C$:
 - i. Restart if $m_{3,7} \geq 1$, $m_{4,8} \geq 1$, $m_{5,9} \geq 1$ or $m_{6,10} \geq 1$.
 - ii. Pick a uniform random $C_{m,n,o,p}$ switching S' on G' .
 - iii. Restart (**f-reject**) with probability $1 - \frac{f_{C_{m,n,o,p}}(G')}{\bar{f}_{C_{m,n,o,p}}(\mathbf{W}')}.$
 - iv. Rewire the edges according to S' and let G'' denote the result.
 - v. Restart (**b-reject**) with probability $1 - \frac{\underline{b}_{C_{m,n,o,p}}(\mathbf{W}')}{\underline{b}_{C_{m,n,o,p}}(G'', v_1 \dots v_{10})}$.
 - vi. Set $G' \leftarrow G''$.
 - (f) Restart (**b-reject**) with probability $1 - \frac{\underline{b}_{TD_0}(\mathbf{W}'; 0) \underline{b}_{TD_0}(\mathbf{W}'; 1)}{\underline{b}_{TD_0}(G', v_1 v_2; 0) \underline{b}_{TD_0}(G', v_1 \dots v_6; 1)}$.
8. Set $G \leftarrow G'$.

It remains to define the quantities for the f- and b-rejection steps. For the f-rejection in step 4, define $f_{TD_1}(G)$ and $f_{TD_0}(G)$ as the number of TD_1 and TD_0 switchings which can be performed on G , respectively. The upper bounds are

$$\bar{f}_{TD_1}(\mathbf{W}) = M^2 T^3, \quad \bar{f}_{TD_0}(\mathbf{W}) = M^4 T^6.$$

For the b-rejections in steps 6c and 7c, define $b_{TD_1}(G', v_1 \dots v_6; 2)$ as the number of timestamps $t_2, t_3 \in [1, T]$ such that $(\{v_3, v_5\}, t_2)$, $(\{v_4, v_6\}, t_3)$ do not exist in G' and $b_{TD_0}(G', v_1 \dots v_{10}; 2)$ as the number of timestamps $t_2, t_3, t_4, t_5 \in [1, T]$ such that $(\{v_3, v_7\}, t_2)$, $(\{v_4, v_8\}, t_3)$, $(\{v_5, v_9\}, t_4)$, $(\{v_6, v_{10}\}, t_5)$ do not exist in G' . The lower bounds are

$$\underline{b}_{TD_1}(\mathbf{W}'; 2) = (T - (\mu - 1))^2, \quad \underline{b}_{TD_0}(\mathbf{W}'; 2) = (T - (\mu - 1))^4.$$

For the b-rejections in steps 6f and 7f, define $b_{TD_1}(G', v_1 \dots v_4; 1)$ as the number of simple edges $(\{v_5, v_6\}, t_6)$ in G' such that v_5, v_6 are distinct from v_1, v_2, v_3, v_4 and $b_{TD_0}(G', v_1 \dots v_6; 1)$ as the number of distinct simple edges $(\{v_7, v_8\}, t_{10})$, $(\{v_9, v_{10}\}, t_{11})$ in G' such that v_7, \dots, v_{10} are distinct from v_1, \dots, v_6 . Then, define $b_{TD_1}(G', v_1 v_2; 0)$ as the number of simple edges $(\{v_1, v_3\}, t_4)$, $(\{v_2, v_4\}, t_5)$ incident at v_1 and v_2 in G' and $\underline{b}_{TD_0}(G', v_1 v_2; 0)$ as the number of distinct simple edges $(\{v_1, v_3\}, t_4)$, $(\{v_2, v_4\}, t_5)$, $(\{v_1, v_5\}, t_6)$, $(\{v_2, v_6\}, t_7)$ incident at v_1 and v_2 in G' . The lower bounds are

$$\begin{aligned} \underline{b}_{TD_1}(\mathbf{W}'; 1) &= M - 4B_D - 4\Delta, & \underline{b}_{TD_0}(\mathbf{W}'; 1) &= (M - 4B_D - 4\Delta)^2, \\ \underline{b}_{TD_1}(\mathbf{W}'; 0) &= k_1 k_2, & \underline{b}_{TD_0}(\mathbf{W}'; 0) &= k_1 (k_1 - 1) k_2 (k_2 - 1). \end{aligned}$$

For the f-rejections in steps 6*ei* and 7*ei*, define $f_{B_{m,n}}(G')$, $f_{C_{m,n,o,p}}(G')$ as the number of $B_{m,n}$, $C_{m,n,o,p}$ switchings which can be performed on G' , respectively. In addition, define

$$\bar{f}_{B_{m,n}}(\mathbf{W}') = (\Delta T)^{2(m+n)}, \quad \bar{f}_{C_{m,n,o,p}}(\mathbf{W}') = (\Delta T)^{2(m+n+o+p)}.$$

For the b-rejections in steps 6*ev* and 7*ev*, define the quantities $b_{B_{m,n}}(G'', v_1 \dots v_6)$ and $b_{C_{m,n,o,p}}(G'', v_1 \dots v_{10})$ as the number of $B_{m,n}$ and $C_{m,n,o,p}$ switchings which can produce the graph G'' , respectively. The corresponding lower bounds are

$$\begin{aligned} \underline{b}_{B_{m,n}}(\mathbf{W}') &= ((M - 4B_D - 4(m+n+3)\Delta)(T - (\Delta - 1))^2)^{m+n}, \\ \underline{b}_{C_{m,n,o,p}}(\mathbf{W}') &= ((M - 4B_D - 4(m+n+o+p+5)\Delta)(T - (\Delta - 1))^2)^{m+n+o+p}. \end{aligned}$$

Once Stage 2 ends, the final graph is simple and can be output.

References

1. Arman, A., Gao, P., Wormald, N.C.: Fast uniform generation of random graphs with given degree sequences. *Random Struct. Algorithms* **59**(3), 291–314 (2021)
2. Barabási, A.L.: The origin of bursts and heavy tails in human dynamics. *Nature* **435**(7039), 207–211 (2005)
3. Bender, E.A., Canfield, E.R.: The asymptotic number of labeled graphs with given degree sequences. *J. Comb. Theory, Ser. A* **24**(3), 296–307 (1978)
4. Blitzstein, J.K., Diaconis, P.: A sequential importance sampling algorithm for generating random graphs with prescribed degrees. *Internet Math.* **6**(4), 489–522 (2011)
5. Bollobás, B.: A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *Eur. J. Comb.* **1**(4), 311–316 (1980)
6. Casteigts, A., Flocchini, P., Quattrociocchi, W., Santoro, N.: Time-varying graphs and dynamic networks. *Int. J. Parallel Emergent Distrib. Syst.* **27**(5), 387–408 (2012)
7. Chung, F., Lu, L.: Connected components in random graphs with given expected degree sequences. *Ann. Comb.* **6**(2), 125–145 (2002)
8. Erdős, P.L., Greenhill, C.S., Mezei, T.R., Miklós, I., Soltész, D., Soukup, L.: The mixing time of switch Markov chains: a unified approach. *Eur. J. Comb.* **99**, 103421 (2022)
9. Gao, P., Wormald, N.C.: Uniform generation of random regular graphs. *SIAM J. Comput.* **46**(4), 1395–1427 (2017)
10. Gao, P., Wormald, N.C.: Uniform generation of random graphs with power-law degree sequences. In: *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018*, pp. 1741–1758. SIAM (2018)
11. Gotelli, N.J., Graves, G.R.: *Null Models in Ecology*. Smithsonian Institution (1996)
12. Greenhill, C.: Generating graphs randomly. *Surv. Comb.* **2021**, 133–186 (2021)
13. Harder, U., Paczuski, M.: Correlated dynamics in human printing behavior. *Phys. A* **361**(1), 329–336 (2006)
14. Holme, P., Saramäki, J.: Temporal networks. *Phys. Rep.* **519**(3), 97–125 (2012)

15. Jerrum, M., Sinclair, A.: Fast uniform generation of regular graphs. *Theor. Comput. Sci.* **73**(1), 91–100 (1990)
16. McKay, B.D., Wormald, N.C.: Uniform generation of random regular graphs of moderate degree. *J. Algorithms* **11**(1), 52–67 (1990)
17. Sinclair, A., Jerrum, M.: Approximate counting, uniform generation and rapidly mixing Markov chains. *Inf. Comput.* **82**(1), 93–133 (1989)
18. Wormald, N.C.: Generating random regular graphs. *J. Algorithms* **5**(2), 247–280 (1984)



A Multi-order Adaptive Network Model for Pathways of DNA Methylation and Its Effects in Individuals Developing Post-traumatic Stress Disorder

Iva Gunjača^(✉), Natalie Samhan, and Jan Treur

Department of Computer Science, Social AI Group, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

gunjaca.iva@gmail.com, j.treur@vu.nl

Abstract. In this paper, an integrative fifth-order biological and mental network model is introduced, to demonstrate epigenetics effects in Post-Traumatic Stress Disorder (PTSD), using an adaptive network modeling approach based on temporal-casual networks. The network model assesses DNA methylation dysregulation, on glucocorticoid receptors, and side effects of glucocorticoid receptor hyperreactivity to glucocorticoids in fear processing of individuals with post-traumatic stress disorder. Furthermore, a form of therapy is explored on differently methylated genes. Genes NR3C1 and FKBP5, and their expression were chosen for the demonstration, furthermore, administering MDMA was chosen as a form of therapy. Several simulations were run to analyze and illustrate behavior of the network model. Behaviors for both genes and their expression were tested together and individually.

Keywords: Epigenetics · post-traumatic stress disorder · DNA methylation · temporal-casual network · fifth-order adaptive network model

1 Introduction

Traumatic events are something most people will experience during their lifetime. After such an event occurs, it is normal to experience a wide range of emotions, however for some individuals these emotions persist [8]. According to the National Institute of Public Health, a Post-Traumatic Stress Disorder (PTSD) is characterized as having ‘at least one re-experiencing symptom, at least one avoidance symptom, at least two arousal and reactivity symptoms and at least two cognition and mood symptoms for the duration of at least one month’ [8]. About every 7 out of 100 people in the general population will develop PTSD at some point in their lifetime [4], while some people recover within 6 months, symptoms may persist for several years without proper treatment [8].

A flashback is categorized as a re-experiencing symptom, reliving the traumatic event, this can include physical symptoms, such as a racing heart or profound sweating. Flashback can be triggered by anything that might remind an individual of the traumatic

event [8]. Individuals of any age, sex, or ethnographic background can develop PTSD. There are several factors that seem to determine if an individual will develop post-traumatic stress disorder after a traumatic event. Some of which are already present before or while a traumatic event is happening, while others only are introduced after a traumatic event has occurred [8].

Recently there has been generated a mounting amount of evidence that epigenetics play a critical role in the development of PTSD. From all epigenetic changes, DNA methylation has been researched the most in individuals with PTSD [4]. DNA methylation is a transfer of a methyl group onto the C5 position of the cytosine to form 5-methylcytosine. This epigenetic mechanism regulates gene expression. For the sake of simplicity, the relationship between DNA methylation and gene expression is assumed as follows: the higher the methylation level (of the regulatory regions of the gene), the lower the expression of said gene is [5].

In this paper, a fifth-order adaptive integrative biological and mental network model was created as a representation of epigenetic DNA methylation changes and its effects on gene NR3C1 expression, and consequences of said expression. PTSD patients are often categorized as having low cortisol levels, which is a direct symptom of increased activity of glucocorticoid receptors. Cortisol is an important hormone in the human body that regulates a variety of systems [21], this model focuses on the effect of cortisol on fear processing. In the paper, a scenario of PTSD development under prolonged chronic stress is introduced, followed with administration of 3,4-Methylenedioxy methamphetamine (MDMA) as a form of therapy. A description of the modeling approach and the software for simulating the model can be found in [10].

2 Background Information on DNA Methylation and Its Effects on Individuals with Post-traumatic Stress Disorder

With strongly increasing interest and developments in epigenetics in recent years, new evidence has emerged that epigenetic factors have a critical role in post-traumatic stress disorder [4]. DNA methylation dysregulation has been observed on two biological processes: the immune system and the stress response [4]. In the current paper, the two most researched genes, NR3C1 and FKBP5 [16], and their expression in individuals with PTSD were chosen to analyze and demonstrate epigenetic changes and their effects. NR3C1 and FKBP5 both have a role in stress response regulation, namely on the glucocorticoid receptors reactivity to glucocorticoids [4, 16]. Glucocorticoid receptor deregulation has been observed in individuals with PTSD. Hyperreactivity of glucocorticoid receptors is considered a main cause of lower cortisol levels that have been observed in individuals with PTSD [17].

For the NR3C1 gene, which encodes glucocorticoid receptors, decreased methylation levels have been observed in individuals with PTSD compared with individuals without PTSD. However, increased methylation on NR3C1 promoter regions has also been observed in individuals with PTSD [4]. For the model, a decrease in methylation on NR3C1 was addressed, as more research can be found on decreased methylation than on increased.

FKBP5 encodes for the FKBP51 protein, a co-chaperone of the glucocorticoid receptor (GR) that regulates GR sensitivity by creating an intracellular ultra-short negative feedback loop [4]. Increased methylation and under-expression of FKBP5 has been observed in individuals with PTSD [18]. However, there have been findings that FKBP5 can serve as a measure of biological cortisol activity [9]. Low cortisol levels, from demethylated NR3C1 gene, could be the main cause of under-expression of FKBP5 [19]. This theory has been taken as point of departure for the developed adaptive network model.

As cortisol has been observed to induce demethylation [19], to explain occurrence of deregulation of methylation levels on NR3C1, cortisol induced demethylation is assumed in the model. DNA repair system methylates the genes as cortisol demethylates. This system was chosen as it has been observed in *Escherichia coli* that when a methylating agent is introduced, DNA repair activities occur and access the damage [7]. These activities have been inherited by humans from bacteria through evolution [7]. Cortisol has been observed to strengthen brain connections in fear processing regions [20]. Because of lowered cortisol levels, there is a possibility that individuals are unable to process the traumatic memory properly and post-traumatic stress disorder develops. Furthermore, it has been observed that cortisol promotes the cognitive control of highly intensive negative emotions [20]. MDMA has been observed as a methylator in cardiac gene expression [27] and in individuals with PTSD on gene NR3C1 [28]. Because of that, it was chosen as a therapy.

3 Methods

A network modeling approach based on temporal-casual networks as described in [10] is used to model DNA methylation and its effects in individuals with PTSD. The general approach to creating these types of network models is as follows. Firstly, based on findings in literature a conceptual model is drawn, a conceptual model can be seen in Fig. 1. Secondly, role matrixes representing models' characteristics are filled out, these role matrixes specify the following network characteristics:

Connectivity. Each incoming connection of a state Y , from a state X has a *connection weight value* $\omega_{X,Y}$ representing the strength of the connection.

Aggregation. For each state a *combination function* $\mathbf{c}_Y(\cdot)$ is chosen to combine the causal impacts state Y receives from other states

Timing. For each state Y a *speed factor* η_Y is used to represent how fast state Y is changing upon causal impact.

Based on the above network characteristics, the difference equation used to compute at time t the impact at later time $t + \Delta t$ on state Y for all its incoming connections is:

$$Y(t + \Delta t) = Y(t) + \eta_Y [\mathbf{c}_Y(\omega_{X_1,Y} X_1(t), \dots, \omega_{X_k,Y} X_k(t)) - Y(t)] \Delta t \quad (1)$$

Here, the X_i are the states from which state Y gets incoming connections. Adaptive-ness of the network model is achieved by making some of these network characteristics adaptive. Instead of a number in the role matrix, a reference to another state (a so-called

self-model state) is written that represents that characteristic. This is called first-order adaptation. If a state for first-order adaptation uses an adaptive network characteristic itself too, a state that represents that characteristic would be characterized as second-order and so on. More about this approach can be found in [10]. The software environment [24] used to simulate the model is described in [10] as well. The software contains a library of around 70 different combination functions that can be used to define each state’s behavior upon incoming causal impact.

4 The Introduced Integrative Adaptive Network Model

The network model introduced here is an integrative fifth-order adaptive network model, integrating a biological network model and a mental network model. Exact names and descriptions of each state can be found in Tables 2, 3 and 4. For a picture of the model, see Fig. 1. For the network model four combination functions were used to describe its behavior, see Table 1.

Table 1. Combination Functions used in the adaptive network model

Function	Notation	Formula	Parameters
Alogistic advanced logistic sum	alogistic $_{\sigma, \tau}(V_1, \dots, V_k)$	$\left[\frac{1}{1 + e^{-\sigma(V_1 + \dots + V_k - \tau)}} - \frac{1}{1 + e^{\sigma\tau}} \right] (1 + e^{-\sigma\tau})$	steepness σ threshold τ
Hebbian Learning	hebb $_{\mu}(V_1, \dots, V_k)$	$V_1 V_2 (1 - V_3) + \mu V_3$	persistence factor μ
Stepmod	stepmod $_{\delta, \tau}(V_1, \dots, V_k)$	time t 0 if $\text{mod}(t, \rho) < \delta$ else 1	repeated time duration ρ tipping point δ
Steponce	steponce $_{\alpha, \beta}(V_1, \dots, V_k)$	time t 1 if $\alpha \leq t \leq \beta$ else 0	start time α end time β

The **advanced logistic sum** is used for most of the states in the model. It was chosen as it has an appropriate range between 0 and 1: it is assumed that activation values of states cannot go above one or below zero, furthermore with the function’s parameter being a threshold of the state, it was a good match to express glucocorticoid receptors sensitivity and FKBP5 expressivity in one. Finally, two models that are adapted for this situation were already using advanced logistic sum for most of the states. **Hebbian learning** was used in adaptive mental network models more often, for more details on its use in developing flashbacks, see [23]. **Step-mod** activates a state repeatedly. It was chosen as a function for a trigger state, as triggers occur frequently. **Step-once** has been chosen as a combinational function for state ss_{t1} , which represents sensor of a traumatic event, furthermore it is also used for therapy state. It was chosen as the state of this function occurs only once for a fixed amount of time. All these functions are a part of the software environment that can be found at [24].

Pathway for Cortisol Production. When pathogens encounter the immune system or an injury occurs, state X_7 (see Table 2) is activated. It represents the infection or injury of the body; in this model it is a constant as it is assumed that the immune system is always in contact with pathogens and this state is used to activate the whole biological network model. Infection (state X_7) affects state X_5 which represents the immune system, when the immune system is active it causes inflammation (state X_6) [1]. This inflammation causes stress to the body (state X_9) [2], and body stress triggers the HPA axis [29], represented by three states X_1 or the hypothalamus, X_2 or the pituitary gland and X_3 or the adrenal cortex. The hypothalamus (state X_1) secretes corticotrophin-releasing hormone which acts as a signal to the pituitary gland (state X_2). In response the pituitary gland produces adrenocorticotrophin hormone which in turn signals the adrenal cortex (state X_3) to start the production of cortisol (state X_4) [29]. Cortisol in return suppresses the immune system [30]. Furthermore, the hypothalamus and pituitary gland contain glucocorticoid receptors (state X_8). When cortisol levels in the body are too high, the glucocorticoid receptors are triggered and suppress the activity of the hypothalamus and pituitary gland [3].

Pathway for Gene Expression. Higher-order states (levels one to five) of the biological model represent the expression of genes NR3C1 (on the left in Fig. 1) and FKBP5 (on the right in Fig. 1). NR3C1 encodes the glucocorticoid receptor and FKBP5 encodes the steroid receptor chaperone FK506 binding protein 51. FKBP51 is a co-chaperone of the glucocorticoid receptor that regulates glucocorticoid receptor sensitivity [4]. To represent the expression of NR3C1 gene, weight from cortisol (state X_4) to glucocorticoid receptor (state X_8) is used, as more NR3C1 gene is expressed the reaction of the glucocorticoid receptor to cortisol is higher. Furthermore, to represent expression of FKBP5, the threshold of the glucocorticoid receptor is used, as the more FKBP5 is expressed the lower the sensitivity of the glucocorticoid receptor becomes [4]. Cortisol induced transcription [6], is represented via positive connection from cortisol to states X_{58} and X_{59} . To represent the effects of epigenetics on gene expression, and the process of transcription and translation, the generic structure from [31] was used as a template and applied to the specific genes considered here.

Pathway for Methylation and Demethylation. State X_{57} and state X_{59} represent the methylation level on their prospective genes. As methylation downregulates the gene expression [5], these states produce a negative effect on states X_{58} and X_{60} , respectively. Glucocorticoid induced demethylation is described as negative connection on states X_{57} and X_{59} , and as demethylation occurs DNA methylation repair system is activated [7]. This relationship is simplified in the model, and it is represented as a positive connection from cortisol to the DNA repair system. The DNA repair system evens out the damage done by cortisol, and this is represented as a positive connection from the DNA repair system to states X_{57} and X_{59} .

Mental Network Model for Flashbacks and Fear Processing. For the mental model of fear processing and development of flashbacks, the model [23] was adopted. A few changes were made to the model to tailor it to this particular model. In the original mental model, some form of psychotherapy is referenced as therapy where the individual learns to control the emotional response to the traumatic event, in our case alternatively

MDMA is used as therapy to methylate the gene that is demethylated, and in return increase cortisol levels. As cortisol is important in strengthening the connections in the brain for fear processing, and those connections are weakened in individuals with PTSD. Because of this, no therapy states ss_{th} and srs_{th} from [23] were included. Furthermore, for the sake of simplicity in contrast to [23] the combination function used for adaptive weights $W_{psb,csb}$ and $W_{fsb,csb}$ was the advanced logistic sum function, and their speeds of adaptation were kept static. For detailed description of the mental model, see [23].

Interaction Between Biological and Mental Network Model. Firstly, the sensory states of the trauma $sste1$, $sste2$, $sste3$ cause chronic stress (state X11) to occur, activating chronic stress, which directly increases the body stress. Furthermore, $srstr$, sensory representation state for trigger causes a small increase in body stress. Secondly, to represent how cortisol strengthens these connections a positive weight is drawn from cortisol (state X4) to these connections. Heightened inflammation levels have been observed in individuals with PTSD [26]. One of points where inflammation levels have been shown to increase is amygdala [26]. This inflammation of amygdala has been speculated to contribute to reduced emotion regulation and fear processing [26]. To describe

Table 2. Base level states and their explanation

Nr	Name	Explanation
X ₁	Hypothalamus	Activity of hypothalamus
X ₂	Pituitary	Activity of pituitary gland
X ₃	Adrenal cortex	Activity of adrenal cortex
X ₄	Glucocorticoids	The level of glucocorticoids in the body
X ₅	Immune System	Activity of the immune system
X ₆	Inflammation	The level of inflammation in the body
X ₇	Infection	Presence of pathogens or injury
X ₈	GR	Reactivity of the glucocorticoid receptors
X ₉	Body Stress	Level of body stress
X ₁₀	DNA repair system	Activity of the DNA repair system
X ₁₁	Chronic stress	Presence of chronic stress
X ₁₂	$sSte1$	Sensor state for traumatic event phase 1: observation te1 (traumatic event)
X ₁₃	$sSte2$	Sensor state for traumatic event phase 2: observation of action te2 (traumatic event)
X ₁₄	$sSte3$	Sensor state for traumatic event phase 3: observation of effect te3 (traumatic event)
X ₁₅	$sStr$	Sensor state for trigger tr for the traumatic event sequence te
X ₁₆	$sFSte1$	Sensory representation state for traumatic event phase 1: observation te1
X ₁₇	$sFSte2$	Sensory representation state for traumatic event phase 2: action te2
X ₁₈	$sFSte3$	Sensory representation state for traumatic event phase 3: effect te3
X ₁₉	$sFStr$	Sensory representation state for trigger tr for traumatic event sequence
X ₂₀	$aSte$	Awareness state for traumatic event te
X ₂₁	psb	Preparation for emotional response b
X ₂₂	fsb	Feeling state for emotional response b / Amygdala
X ₂₃	csb	Control state for emotional response b
X ₂₄	bsb_{te}	Belief that emotional response b is from traumatic event te
X ₂₅	esb	Bodily expressed emotional response b
X ₂₆	esb_{te}	Expressing that emotional response b is from traumatic event te

this observation a positive connection is drawn from inflammation state (state X6) to feeling state fsb.

Table 3. First-order adaptation level states and their explanation

Nr	Name	Explanation
X ₂₇	con _{CHR}	Context state for enabling CHR production
X ₂₈	W _{Hypothalamus, Pituitary}	First-order self-model state for the weights of the base level connections from hypothalamus to pituitary, on a causal pathway from context state con _{CHR}
X ₂₉	con _{ACTH}	Context state for enabling ACTH production
X ₃₀	W _{Pituitary, Adrenal cortex}	First-order self-model state for the weight of the base level connections from pituitary to adrenal cortex, on a causal pathway from context state con _{ACTH}
X ₃₁	con _{reactivity of GR on glucocorticoids}	Context state for reactivity for glucocorticoid receptors on glucocorticoids
X ₃₂	W _{glucocorticoids, GR}	First-order self-model state for the weights of the base level connections from glucocorticoids to glucocorticoid receptors; on a causal pathway from context state con _{reactivity of glucocorticoids on GR}
X ₃₃	con _{FK506 binding protein 51}	Context state for FK506 binding protein 51 production
X ₃₄	T _{GR}	First-order self-model state for the threshold of the combination function of GR; on a causal pathway with a connection from context state con _{FK506 binding protein 51}
X ₃₅	W _{srs_{te1}, srs_{te2}}	First-order self-model state for weight of the connection from srs _{te1} to srs _{te2} for imprinting the traumatic sequence
X ₃₆	W _{srs_{te2}, srs_{te3}}	First-order self-model state for weight of the connection from srs _{te2} to srs _{te3} for imprinting the traumatic sequence
X ₃₇	W _{srs_{tr}, srs_{te1}}	First-order self-model state for weight of the connection from srs _{tr} to srs _{te1} for sensory preconditioning to link trigger tr to the traumatic event
X ₃₈	W _{ps_b, cs_b}	First-order self-model state for the weight of the connection ps _b to cs _b for strength of regulation
X ₃₉	W _{fs_b, cs_b}	First-order self-model state for the weight of the connection fs _b to cs _b for strength of regulation

5 Simulation Results

To demonstrate and test the network model's behavior some of the simulations run in MATLAB are discussed. In Scenario 1 a post-traumatic stress disorder is developed under prolonged exposure to a stressful event. Scenario 4 then follows with administration of MDMA as a form of therapy to properly methylate the demethylated gene. Scenarios 2 and 3 were run to determine if the model behaves differently when only one gene influences models' behavior and if there is an interaction, inside of the model, between expression of FKBP5 and NR3C1. Difference in FKBP5 expression was expected.

Scenario 1: Development of Post-Traumatic Stress Disorder. From the beginning of the simulation the infection (state X7) is a constant that affects the model's process. The trigger for a fear response occurs from 100 to 200 and after that regularly recurs in intervals from 300 to 400, from 500 to 600, et cetera. A traumatic event is introduced from 1000 to 1500, which starts the development of post-traumatic stress disorder in the

Table 4. Second- to fifth-order adaptation level states and their explanation (levels indicated in different shades like in Fig. 1)

Nr	Name	Explanation
X ₄₀	$W_{\text{COI}}^{\text{CHR}}, W_{\text{Hypothalamus, Pituitary}}$	Self-model state for the weight of the first self-modelling level connection from context state con_{CHR} (context for W -states) to $W_{\text{Hypothalamus, Pituitary}}$ for connection from hypothalamus to pituitary gland
X ₄₁	$W_{\text{COI}}^{\text{ACTH}}, W_{\text{Pituitary, Adrenal cortex}}$	Self-model state for the weight of the first self-modelling level connection from context state con_{ACTH} to $W_{\text{Pituitary, Adrenal cortex}}$ for connection from pituitary gland to adrenal cortex
X ₄₂	$\text{COI}^{\text{NR3C1enzymes}}$	Context state for enabling enzyme production for gene NR3C1
X ₄₃	$W_{\text{COI}}^{\text{reactivity of GR on glucocorticoids}}, W_{\text{glucocorticoids, GR}}$	Self-model state for the weight of the first self-modelling level connection from context state $\text{con}_{\text{reactivity of GR on glucocorticoids}}$ to $W_{\text{glucocorticoids, GR}}$ for connection from glucocorticoids to glucocorticoid receptors; on a causal pathway with a connection from $\text{COI}^{\text{NR3C1enzymes}}$
X ₄₄	$\text{COI}^{\text{FKBP5enzymes}}$	Context state for enabling enzyme production for gene FKBP5
X ₄₅	$W_{\text{COI}}^{\text{FK506 binding protein 51}}, T$	Self-model state for the threshold of the first self-modelling level connection from context state $\text{con}_{\text{FK506 binding protein 51}}$ (context for T -states) to T_{GR} for sensitivity of glucocorticoid receptors; on a causal pathway with a connection from $\text{COI}^{\text{FKBP5enzymes}}$
X ₄₆	$H_{\text{W}}^{\text{srs1e1, srs1e2}}$	Control state for adaptation speed for weight of connection from srs_{1e1} to srs_{1e2}
X ₄₇	$H_{\text{W}}^{\text{srs1e2, srs1e3}}$	Control state for adaptation speed for weight of connection from srs_{1e2} to srs_{1e3}
X ₄₈	$H_{\text{W}}^{\text{srs1r, srs1e1}}$	Control state for adaptation speed for weight of connection from srs_{1r} to srs_{1e1}
X ₄₉	$\text{COI}^{\text{NR3C1mRNA}}$	Context state for enabling mRNA production for gene NR3C1
X ₅₀	$W_{\text{COI}}^{\text{NR3C1enzymes}}, W_{\text{COI}}^{\text{reactivity of GR on glucocorticoids}}, W_{\text{glucocorticoids, GR}}$	Self-model state for the weight of the second-order self-modelling level connection from context state $\text{con}_{\text{NR3C1enzymes}}$ to $W_{\text{COI}}^{\text{reactivity of GR on glucocorticoids}}, W_{\text{glucocorticoids, GR}}$ for connection from glucocorticoids to glucocorticoid receptors; on a causal pathway with a connection from $\text{COI}^{\text{NR3C1mRNA}}$
X ₅₁	$\text{COI}^{\text{FKBP5mRNA}}$	Context state for enabling mRNA production for gene FKBP5
X ₅₂	$W_{\text{COI}}^{\text{FKBP5enzymes}}, W_{\text{COI}}^{\text{FK506 binding protein 51}}, T$	Self-model state for the threshold of the second-order self-modelling level connection from context state $\text{con}_{\text{FKBP5enzymes}}$ to $W_{\text{COI}}^{\text{FK506 binding protein 51}}, T$ for sensitivity of glucocorticoid receptors; on a causal pathway with a connection from $\text{COI}^{\text{FKBP5mRNA}}$
X ₅₃	$\text{COI}^{\text{NR3C1DNA}}$	Context state for gene NR3C1
X ₅₄	$W_{\text{COI}}^{\text{NR3C1mRNA}}, W_{\text{COI}}^{\text{NR3C1enzymes}}, W_{\text{COI}}^{\text{reactivity of GR on glucocorticoids}}, W_{\text{glucocorticoids, GR}}$	Self-model state for the weight of the third-order self-modelling level connection from context state $\text{con}_{\text{NR3C1mRNA}}$ to $W_{\text{COI}}^{\text{NR3C1enzymes}}, W_{\text{COI}}^{\text{reactivity of GR on glucocorticoids}}, W_{\text{glucocorticoids, GR}}$ for connection from glucocorticoids to glucocorticoid receptors, on a causal pathway with a connection from $\text{COI}^{\text{NR3C1DNA}}$
X ₅₅	$\text{COI}^{\text{FKBP5DNA}}$	Context state for gene FKBP5
X ₅₆	$W_{\text{COI}}^{\text{FKBP5mRNA}}, W_{\text{COI}}^{\text{FKBP5enzymes}}, W_{\text{COI}}^{\text{FK506 binding protein 51}}, T$	Self-model state for the threshold of the third-order self-modelling level connection from context state $\text{con}_{\text{FKBP5mRNA}}$ to $W_{\text{COI}}^{\text{FKBP5enzymes}}, W_{\text{COI}}^{\text{FK506 binding protein 51}}, T$ for sensitivity of glucocorticoid receptors; on a causal pathway with a connection from $\text{COI}^{\text{FKBP5DNA}}$
X ₅₇	$\text{COI}^{\text{NR3C1Methylation}}$	Context state suppressing NR3C1/ Representation of methylation level on NR3C1
X ₅₈	$W_{\text{COI}}^{\text{NR3C1DNA}}, W_{\text{COI}}^{\text{NR3C1mRNA}}, W_{\text{COI}}^{\text{NR3C1enzymes}}, W_{\text{COI}}^{\text{reactivity of GR on glucocorticoids}}, W_{\text{glucocorticoids, GR}}$	Self-model state for the weight of the fourth self-modelling level connection from context state $\text{con}_{\text{NR3C1DNA}}$ to $W_{\text{COI}}^{\text{NR3C1mRNA}}, W_{\text{COI}}^{\text{NR3C1enzymes}}, W_{\text{COI}}^{\text{reactivity of GR on glucocorticoids}}, W_{\text{glucocorticoids, GR}}$ for connection from glucocorticoids to glucocorticoid receptors; on a causal pathway with a connection from $\text{con}_{\text{NR3C1Methylation}}$
X ₅₉	$\text{COI}^{\text{FKBP5Methylation}}$	Context state for suppressing FKBP5/Representation of methylation levels on FKBP5
X ₆₀	$W_{\text{COI}}^{\text{FKBP5DNA}}, W_{\text{COI}}^{\text{FKBP5mRNA}}, W_{\text{COI}}^{\text{FKBP5enzymes}}, W_{\text{COI}}^{\text{FK506 binding protein 51}}, T$	Self-model state for the threshold of the fourth-order self-modelling level connection from context state $\text{con}_{\text{FKBP5DNA}}$ for T_{GR} for sensitivity of glucocorticoid receptors; on a pathway with a connection from $\text{COI}^{\text{FKBP5Methylation}}$
X ₆₁	MDMA	Presence of MDMA in the body

mental and biological network model. For demonstration of this simulation, the MDMA state has been 0 for the duration of whole simulation. This simulation can then serve as a baseline to compare the effects of administering MDMA.

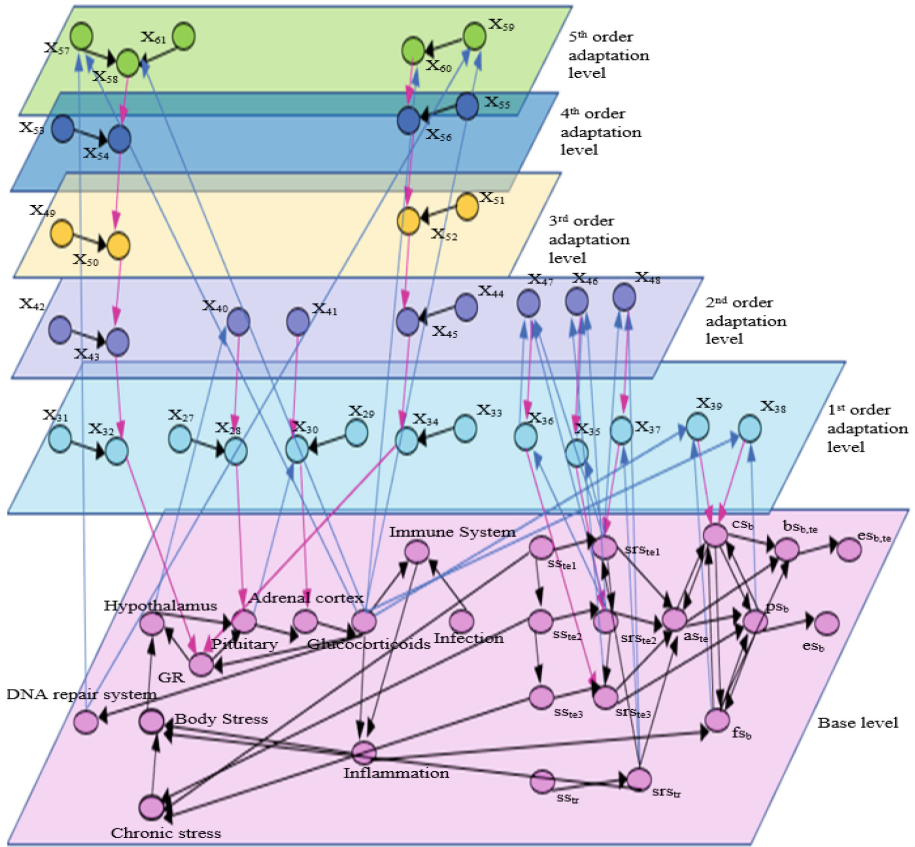


Fig. 1. 3D picture of the adaptive network model. For explanations of the states of the different levels, see Tables 2, 3 to 4.

Scenario 2: Development of Post-Traumatic Stress Disorder (only NR3C1). For this scenario, an identical process is applied as in simulation one, with one major difference, the effects of FKPB5 have been turned off, by making the glucocorticoid receptor state have a fixed excitability threshold value instead of an adaptive one.

Scenario 3: Development of Post-Traumatic Stress Disorder (only FKPB5). For this scenario, an identical process is applied as in simulation one, with one other major difference, the effects of NR3C1 have been turned off, by making the weight characteristic from cortisol to the glucocorticoid receptor a fixed number instead of an adaptive characteristic.

Scenario 4: Symptom Reduction by Administration of MDMA. Here, in addition to the development of PTSD, therapy in the form of MDMA administration occurs from 2500 to 2700.

5.1 Scenario 1: Development of Post-Traumatic Stress Disorder

While chronic stress is in effect, we can observe an increase in cortisol (purple line in Fig. 2) and its production system. Because of the high levels of cortisol, GR has an increase in activity (dark blue line in Fig. 2). While the immune system (green line in Fig. 2) and its product, inflammation (blue line in Fig. 2), is decreasing. This behavior accurately represents a person under the influence of a traumatic event.

Glucocorticoids demethylate both genes, and the DNA repair system has sufficient level of activation to be able to repair the damage done by cortisol. However, if damage done by glucocorticoids becomes too high for the DNA repair system to fix, on gene NR3C1 the damage becomes permanent. Because of permanent demethylation on NR3C1, the gene is overexpressed, GRs react more aggressively to cortisol which ultimately leads to lower cortisol levels being observed in individuals with PTSD. As cortisol is now lower than usual, the natural balance of methylation and demethylation on gene FKBP5 is disrupted, and FKBP5 is silenced. This leads to high sensitivity of GRs, and even higher reactivity. This behavior can be observed in Figs. 6, 7 and 9 in the appendix [32].

Regarding the mental network model, after development of PTSD, all the states have a higher reactivity as every time a trigger occurs a mental video of a traumatic event is played which creates a bigger reaction, than in a healthy person would occur. The weights are higher than in a healthy person, however that is also expected as reactions are also higher. This behavior can be observed in Figs. 5, 10 and 11 in the appendix [32].

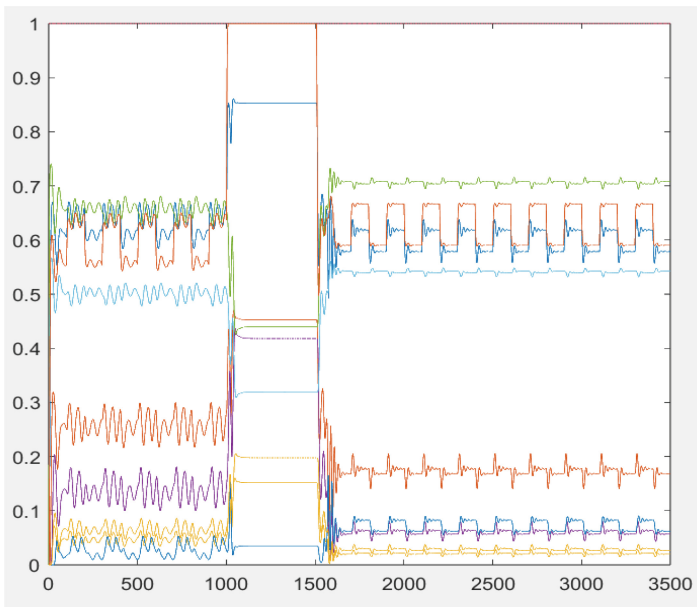


Fig. 2. Simulation results for Scenario 1 (biological network – base level)

5.2 Scenario 2: Development of Post-Traumatic Stress Disorder (only NR3C1)

When FKPB5 no longer effects the pathways in the network model, methylation at NR3C1 returns to normal after chronic stress has passed, in turn the whole biological side of the model continues acts as in a healthy person after chronic stress has passed. After chronic stress has passed, values of the mental network model act as after symptom reduction from administration of MDMA. This behavior can be seen in Figs. 21 and 22 in the Appendix [32].

5.3 Scenario 3: Development of Post-Traumatic Stress Disorder (only FKPB5)

When NR3C1 no longer affects the model, values of the model act more dynamic and FKPB5 values don't drop as much as in scenario one, during chronic stress. Methylation on gene FKPB5 returns to homeostatic state after chronic stress has passed, and the biological network model returns to normal after chronic stress has passed. The mental network model acts as after symptom reduction from administration of MDMA. This behavior can be seen in Figs. 23 and 24 in the Appendix [32].

5.4 Scenario 4: Symptom Reduction by Administration of MDMA

When MDMA is administered, methylation and the rest of the biological network model goes back to its previous values. Control state is increased, and the individual has better control of their emotions, however still not at the level of a healthy person. You can observe this behavior in Figs. 12 to 20 in the Appendix [32].

6 Discussion

The presented results give an insight into the development of post-traumatic stress disorder and the underlying epigenetic processes that contribute to its manifestation. Chronic stress has already been associated with alternations to cortisol levels and its regulatory system [11]. The observed increase in cortisol during a traumatic event, and in turn heightened activity of glucocorticoid receptors, DNA repair system and reduced inflammation, are a typical reaction to a potential treat [12–14]. This study introduces a computational analysis perspective on how cortisol indirectly impacts gene expression through demethylation, and the DNA repair system through DNA methylation. However, in the case of NR3C1, the damage caused by cortisol surpasses the repair capacity, leading to permanent demethylation. This behavior correlates with previous findings, that in individuals that are exposed to chronic stress, there are higher levels of DNA damage than in healthy individuals [15]. This overexpression of NR3C1 intensifies the GR's reaction to cortisol, contributing to overall lowered cortisol levels in individuals with PTSD. Due to disrupted cortisol levels FKBP5 is silenced, which in return amplifies the sensitivity and reactivity of the glucocorticoid receptor even more. All this was simulated by the introduced adaptive network model.

The observation that removal of FKBP5 from the model results in restoration of methylation levels on gene NR3C1 after exposure to a traumatic event, underscores the

potential therapeutic target of FKBP5 in managing the epigenetic alternations associated with PTSD development. Potential therapeutic effects of administering MDMA for individuals with PTSD were explored as well in this study. The behavior of the network model follows the patterns as expected from the real world. While methylation does increase when individuals are subjected to MDMA therapy, the symptoms of individuals with PTSD are reported to decrease, but not disappear altogether [28]. As PTSD is more complex than just one differently methylated gene.

While exposure therapy is considered the golden standard for dealing with PTSD, dropout rates from the therapy are high, and it is not always successful [28]. There is an increasing need for some kind of pharmaceutical therapy that can be used in parallel with psychotherapy. This paper is describing the effects which MDMA has on the activity of glucocorticoid receptors, and how changes in glucocorticoid receptors behavior affect the individuals fear processing. MDMA seems to do more than just fix one demethylated gene for improvement of PTSD symptoms [28], however, as this model is focused solely on the reactivity of glucocorticoid receptors, other effects were left out. In the future other biological systems can be explored and its effects on the mental network model within the overall network model can be expanded upon. Not all is known on DNA methylation and the biological background of fear processing, both are fields that we have just recently started to properly explore as technology improves. While every part of the model can be referenced to a scientific paper, some of the claims are just theories for now, but with a solid background in the literature. Overall, the approach fits very well in the general perspective on the role of epigenetics in mental disorders described in the recently contributed [33].

In individuals with PTSD methylation changes tend to occur at different genes, and they don't always concern low cortisol. Age at the time of trauma, sex and genetic background seem to determine what kind of physical problems show as symptoms. This information shows that there are different pathways which can lead to the development of PTSD. In this model we were following an individual with lowered levels of cortisol and GR hyperactivity.

In conclusion, the presented results provide an insight in the development of PTSD, and interactions between cortisol, gene expression and emotion regulation. Additionally, the potential of MDMA as a therapy for PTSD is explored, offering hope for restoration of proper emotion regulation.

References

1. Chen, L., et al.: Inflammatory responses and inflammation-associated diseases in organs. *Oncotarget*. **9**(6), 7204–7218 (2017). <https://doi.org/10.18632/oncotarget.23208>. PMID: 29467962; PMCID: PMC5805548
2. Antonangeli, F., Grimsholm, O., Rossi, M.N., Velotti, F.: Editorial: cellular stress and inflammation: how the immune system drives tissue homeostasis. *Front. Immunol.* **12**, 668876 (2021). <https://doi.org/10.3389/fimmu.2021.668876>. PMID: 33828567; PMCID: PMC8021016
3. Ramamoorthy, S., Cidlowski, J.A.: Corticosteroids: mechanisms of action in health and disease. *Rheum Dis. Clin. North Am.* **42**(1), 15–31, vii (2016). <https://doi.org/10.1016/j.rdc.2015.08.002>. PMID: 26611548; PMCID: PMC4662771

4. Morrison, F.G., Miller, M.W., Logue, M.W., Assef, M., Wolf, E.J.: DNA methylation correlates of PTSD: Recent findings and technical challenges. *Prog. Neuropsychopharmacol. Biol. Psychiatry* **90**, 223–234 (2019). <https://doi.org/10.1016/j.pnpbp.2018.11.011>
5. Moore, L., Le, T., Fan, G.: DNA methylation and its basic function. *Neuropsychopharmacol* **38**, 23–38 (2013). <https://doi.org/10.1038/npp.2012.112>
6. Meijsing, S.H.: Mechanisms of glucocorticoid-regulated gene transcription. *Adv. Exp. Med. Biol.* **872**, 59–81 (2015). https://doi.org/10.1007/978-1-4939-2895-8_3. PMID: 26215990
7. Sedgwick, B.: Repairing DNA-methylation damage. *Nat. Rev. Mol. Cell Biol.* **5**, 148–157 (2004). <https://doi.org/10.1038/nrm1312>
8. National Institute of Public Health. <https://www.nimh.nih.gov/health/publications/post-traumatic-stress-disorder-ptsd>
9. Bancos, I., Hatipoglu, B.A., Yuen, K.C.J., Chandramohan, L., Chaudhari, S., Moraitis, A.G.: Evaluation of FKBP5 as a cortisol activity biomarker in patients with ACTH-dependent cushing syndrome. *J. Clin. Transl. Endocrinol.* **24**, 100256 (2021). <https://doi.org/10.1016/j.jcte.2021.100256>
10. Treur, J.: *Network-Oriented Modeling for Adaptive Networks: Designing Higher-Order Adaptive Biological, Mental and Social Network Models*. Springer Nature (2020). <https://doi.org/10.1007/978-3-030-31445-3>
11. Hannibal, K.E., Bishop, M.D.: Chronic stress, cortisol dysfunction, and pain: a psychoneuroendocrine rationale for stress management in pain rehabilitation. *Phys. Ther.* **94**(12), 1816–1825 (2014). <https://doi.org/10.2522/ptj.20130597>
12. Cay, M., et al.: Effect of increase in cortisol level due to stress in healthy young individuals on dynamic and static balance scores. *Northern Clinics Istanbul* **5**(4), 295–301 (2018)
13. Forlenza, M.J., et al.: The effects of stress on DNA repair capacity. *Psychol. Health* **15**(6), 881–891 (2000). <https://doi.org/10.1080/08870440008405589>
14. Dhabhar, F.S.: A hassle a day may keep the pathogens away: the fight-or-flight stress response and the augmentation of immune function. *Integr. Comp. Biol.* **49**(3), 215–236 (2009)
15. Dimitroglou, E., Messini-Nikolaki, N., Doudounakis, S., Tsilimigaki, S., Piperakis, S.: DNA damage in human population affected by chronic psychogenic stress. *Int. J. Hyg. Environ. Health* **206**, 39–44 (2003). <https://doi.org/10.1078/1438-4639-00187>
16. Cao-Lei, L., et al.: A narrative review of the epigenetics of post-traumatic stress disorder and post-traumatic stress disorder treatment. *Front. Psychiatry* **13**, 857087 (2022)
17. Yehuda, R., Seckl, J.: Minireview: stress-related psychiatric disorders with low cortisol levels: a metabolic hypothesis. *Endocrinology* **152**(12), 4496–4503 (2011). <https://doi.org/10.1210/en.2011-1218>
18. Yehuda, R.: Gene expression patterns associated with ... - biological psychiatry. *Biological Psychiatry* (2009). [https://www.biologicalpsychiatryjournal.com/article/S0006-3223\(09\)00394-1/fulltext](https://www.biologicalpsychiatryjournal.com/article/S0006-3223(09)00394-1/fulltext)
19. Yehuda, R., et al.: Epigenetic biomarkers as predictors and correlates of symptom improvement following psychotherapy in combat veterans with PTSD. *Front. Psych.* **4**, 118 (2013). <https://doi.org/10.3389/fpsy.2013.00118>
20. Merz, C.J., Hamacher-Dang, T.C., Stark, R., Wolf, O.T., Hermann, A.: Neural underpinnings of cortisol effects on fear extinction. *Neuropsychopharmacology* **43**(2), 384–392 (2018). <https://doi.org/10.1038/npp.2017.227>
21. Thau, L., Gandhi, J., Sharma, S.: Physiology, cortisol. In: *StatPearls* [Internet]. Treasure Island (FL): StatPearls Publishing; 2023 Jan-. (2022). <https://www.ncbi.nlm.nih.gov/books/NBK538239/>
22. Treur, J.: *Network-Oriented Modeling: Addressing Complexity of Cognitive, Affective and Social Interactions*. Springer Nature (2016). <https://doi.org/10.1007/978-3-319-45213-5>
23. van Ments, L., Treur, J.: Disturbed by Flashbacks: A Controlled Adaptive Network Model Addressing Mental Models for Flashbacks from PTSD. In [25]

24. Treur, J., Van Ments, L. (eds.) *Mental Models and Their Dynamics, Adaptation, and Control. Studies in Systems, Decision and Control*, vol. 394. Springer, Cham. https://doi.org/10.1007/978-3-030-85821-6_5
25. Treur, J.: *Network-Oriented Modeling Software* (2023). <https://www.researchgate.net/publication/368775720>
26. Hori, H., Kim, Y.: Inflammation and post-traumatic stress disorder. *Psychiatry Clin. Neurosci.* **73**(4), 143–153 (2019). <https://doi.org/10.1111/pcn.12820>
27. Koczor, C.A., et al.: Ecstasy (MDMA) alters cardiac gene expression and DNA methylation: implications for circadian rhythm dysfunction in the heart. *Toxicol. Sci. Offic. J. Soc. Toxicol.* **148**(1), 183–191 (2015). <https://doi.org/10.1093/toxsci/kfv170>
28. Lewis, C.R., et al.: Pilot study suggests DNA methylation of the glucocorticoid receptor gene (NR3C1) is associated with MDMA-assisted therapy treatment response for severe PTSD. *Front. Psych.* **14**, 959590 (2023). <https://doi.org/10.3389/fpsy.2023.959590>
29. Dunlavey, C.J.: Introduction to the hypothalamic-pituitary-adrenal axis: healthy and dysregulated stress responses, developmental stress and neurodegeneration. *J. Undergraduate Neurosci. Educ.* June : A Publication of FUN, Faculty for Undergraduate Neuroscience **16**(2), R59–R60 (2018)
30. Morey, J.N., Boggero, I.A., Scott, A.B., Segerstrom, S.C.: Current directions in stress and human immune function. *Curr. Opin. Psychol.* **5**, 13–17 (2015). <https://doi.org/10.1016/j.copsyc.2015.03.007>
31. Kathusing, S., Samhan, N., Treur, J.: Higher-Order Adaptive Dynamical System Modeling of the Role of Epigenetics in Anxiety Disorders. *Cognitive Systems Research*, in press (2023)
32. Appendix: Available as Linked Data at <https://www.researchgate.net/publication/373756621>
33. Nigg, J.T.: Considerations toward an epigenetic and common pathways theory of mental disorder. *J. Psychopathol. Clin. Sci.* **132**(3), 297–313 (2023)



DynamicScore: A Novel Metric for Quantifying Graph Dynamics

Bridonneau Vincent, Guinand Frédéric^(✉), and Pigné Yoann

LITIS Lab, Université Le Havre Normandie, Le Havre, France
{vincent.bridonneau, frederic.guinand, yoann.pigne}@univ-lehavre.fr

Abstract. This study introduces a new metric called “DynamicScore” to evaluate the dynamics of graphs. It can be applied to both vertices and edges. Unlike traditional metrics, DynamicScore not only measures changes in the number of vertices or edges between consecutive time steps, but also takes into account the composition of these sets. To illustrate the possible contributions of this metric, we calculate it for increasing networks of preferential attachment (Barabási-Albert model) and Edge-Markovian graphs. The results improve our understanding of the dynamics inherent in these generated evolving graphs.

Introduction

Dynamic graphs refer to graphs subject to changes along time. Apart from the term ‘dynamic graphs,’ which can be found in [7], the terminology is varied. The most common terms mentioned in the scientific literature include ‘evolving graphs’ [6], ‘dynamic networks’ [10], ‘temporal networks’ [8], ‘time-varying graphs’ [4], and ‘temporal graphs’ [9]. A Dynamic graph can be defined as a sequence of snapshot graphs ordered by a timestamp. Many problems arising in a wide variety of systems have been formulated using dynamic graphs. Among them, as mentioned in [2], the analysis and understanding of complex networks require the design of network growth models and graph evolution mechanisms. The generation process always starts from an initial seed graph G_0 (the initial element of the sequence of snapshot graphs). Then, at each step, a new graph is generated by applying rules to previously generated graphs. A comprehensive description of this process is given in [3]. This new graph is then appended to the sequence, and the process continues until a specified condition is met or results in an infinite number of graphs. The snapshot graph produced at step t is both the current last element of the sequence produced by the generator and a resource element for the generator itself as illustrated on Fig. 1.

Numerous challenges in graph theory have been revisited in the context of dynamic graphs. By introducing the temporal dimension, novel metrics have emerged, and classical properties have been redefined. These include time-respecting paths, reachability, temporal connectivity, and persistent patterns, among others. Nevertheless, it is noteworthy that, to the best of our knowledge, limited research has been dedicated to exploring the intricate relationship

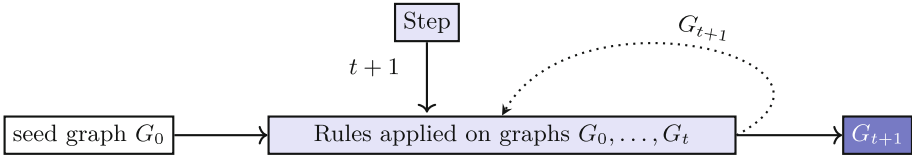


Fig. 1. Synthetic description of Dynamic Graphs Generators

between the generative process and the inherent dynamics of the graph itself. Some prior works presuppose a limited number of changes between two consecutive snapshot graphs. Others characterize the dynamics using terms like ‘slow dynamics’, ‘not frequent changes’, ‘similar consecutive snapshot graphs’ to cite a few. In both scenarios, there exists a clear need for a metric capable of quantifying the dynamism of the graph between two consecutive snapshot graphs. This metric should not only capture changes in the cardinality of vertex and edge sets but also alterations in their composition.

In this work, the DynamicScore metric, coping with both aspects, is presented¹. The metric is implemented for the set of vertices, V-DynamicScore, and for the set of edges, E-DynamicScore. Our main contribution is a novel analysis of two state-of-the-art dynamic graphs generative processes based on this metric: the preferential attachment growing model by Barabási and Albert [1] and the Edge-Markovian Graph model [5]. In the next Section the metric is formally defined and some singular values corresponding to peculiar graph evolutions are presented and discussed. Section 2 is dedicated to the analysis of DynamicScore on graphs generated using the Preferential Attachment growing model. It is shown that the dynamics of the graph decreases as the number of steps increases. Section 3 starts with a description of the Edge-Markovian Graphs Generator (EMGG) and outlines some properties of the generated graphs. Then the analysis of the dynamics of Edge-Markovian graphs is conducted and some results about edge dynamics with respect to the parameters of the model are presented. We conclude this work by introducing two open questions about relationships between Markovian-based dynamic graph evolution and the DynamicScore.

1 DynamicScore

The DynamicScore, which is derived from the Jaccard distance, encompasses several properties that shed light on the nature of a dynamic graph. It effectively captures the degree of dynamics exhibited by the graph, whether it is applied to the vertices or the edges. Notably, DynamicScore emphasizes changes in composition, both at a local level over time between two consecutive steps, and at a global level spanning the entire graph. It is formally defined as follow:

¹ In [3], this metric was referred to as ‘nervousness,’ a translation of a French term that could be misleading in English.

Definition 1. *V-DynamicScore:*

Given a dynamic graph G , such that at time t $G_t = (V_t, E_t)$. We call **V-DynamicScore** at time t and denoted by \mathcal{D}_t^v , the ratio:

$$\mathcal{D}_t^v = \frac{|V_{t+1} \Delta V_t|}{|V_{t+1} \cup V_t|}$$

where $|A|$ denotes the number of edges present in set A . The Δ operator for all set A and B , referred to as $A \Delta B$, is defined as $A \cup B - A \cap B$.

Similarly, for a given dynamic graph the definition of its edges DynamicScore is defined as follow:

Definition 2. *E-DynamicScore:*

Given a dynamic graph G , such that at time t $G_t = (V_t, E_t)$. We call **E-DynamicScore** at time t and denoted by \mathcal{D}_t^e , the ratio:

$$\mathcal{D}_t^e = \frac{|E_{t+1} \Delta E_t|}{|E_{t+1} \cup E_t|}$$

The DynamicScore serves as a similarity metric, enabling comparisons between two consecutive snapshot graphs. A score of 0 indicates that the two graphs are identical, while a score of 1 signifies that they do not share any common vertices. In general, a value close to 0 suggests minimal changes in the graph between two consecutive steps, whereas a value close to 1 implies significant modifications have occurred. It should be noted that graph order and DynamicScore measure two different quantities. For instance, between two consecutive time steps, t and $t + 1$, the value of Vertex-DynamicScore can be equal to 1 while the order of the graph remains the same. This occurs when all the vertices have changed between t and $t + 1$. In the next two sections the analysis will mainly focus on the dynamics of Vertex and E-DynamicScore of the Barabási model as defined in [1] and the EMGG model.

2 Analysis of the Dynamics of the Preferential Attachment Growing Model

2.1 Introduction to the Model

In [1], the generative process is clearly described. For the first part of our analysis, we only focus on the evolution of the number of vertices and on the number of edges. Using our notations the generation of the graph starts with a seed graph $G_0 = (V_0, E_0)$ such that $|V_0| = n_0$ and $0 \leq |E_0| = m_0 \leq \frac{1}{2}n_0(n_0 - 1)$. Note that in the original research article, no information is given about the initial number of edges. At every time step $t + 1$ a new vertex is added and this new vertex is linked to $m (\leq n_0)$ vertices already in V_t . Thus $|V_{t+1}| = |V_t| + 1$ and $|E_{t+1}| = |E_t| + m$.

2.2 DynamicScore

From this it is possible to compute both Vertex and E-DynamicScore. As the number of node inserted in the graph at each step is one, $\mathcal{D}_t^v = \frac{1}{n_0+t+1}$. Moreover, the number of new connections being m and no connection being removed leads to $\mathcal{D}_t^e = \frac{m}{m_0+tm}$. Thus, both the Vertex and the E-DynamicScore are decreasing and tends toward 0 as t tends to infinity.

3 Generator of Edge-Markovian Graphs

This section presents the Edge-Markovian Graphs Generator (EMGG), its formal definition and some of its fundamental properties. In the first part, we present the model and its characteristics. Moving on to the second part, we delve into the general results and explore the relationships between EMGG and the DynamicScore metric. These results unveil a significant connection between the graph's density and the value of DynamicScore, shedding light on their interplay and implications. By examining this relationship, we gain valuable insights into the dynamics of the graph and the quantitative assessment provided by DynamicScore.

3.1 The Model

The Edge-Markovian Graphs Generator (EMGG) is a stochastic process that produces an infinite sequence of static graphs. We denote G_t the graph produced at step t . $G_t = (V_t, E_t)$ where V_t (resp. E_t) represents the set of vertices (resp. edges) at step t .

The EMGG is parameterized by two probabilities, denoted as p and q , along with an initial condition or seed graph, denoted as G_0 . The set of vertices of the graph does not change during the evolution process, so, for all $t > 0$, $V_t = V_0 = n$. Given two vertices u and v , if at step t the edge $(u, v) \in E_t$, the edge is said *present* and *absent* otherwise. The EMGG operates as follows: at each step, all possible edges (present or absent) are examined². The generator determines for each edge if it will remain in the same state (present/absent) in the next snapshot graph or if it will change. The decision is based on two probability parameters: $0 \leq p \leq 1$ and $0 \leq q \leq 1$. The role of p is to define the probability that an edge present at a given step remains present during the next step, while the role of q is to define the probability that an edge absent at a given step remains absent during the next step. This is summarized in the following diagram:

There are several special cases worth noting. When both p and q are set to 0, the generated graphs exhibit a blinking behavior, where edges alternate between present and absent at each step. On the other hand, when both p and q are set to 1, the generated graphs remain static throughout the sequence, with G_t being equal to the initial graph G_0 for all time steps. In the case where $q = 1 - p$, the generating process becomes "time-homogeneous", meaning that the generation of the new graph at each step does not depend on the previous step (Fig. 2).

² there are $n(n-1)/2$ such edges.

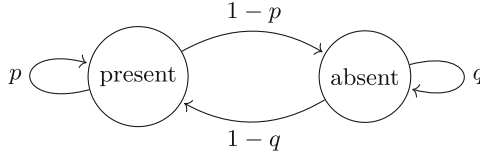


Fig. 2. Description of the states.

Definition 3. An EMGG is parameterized through 4 parameters $n \in \mathbb{N}^*$, p and $q \in [0, 1]$ and an initial configuration G_0 . Instances produced by such a generator are such that:

- for all step t , $|V_t| = n$;
- for pair of vertices $e = (u, v) \in V_t^2$:
 - if $e \in E_t$, then $e \in E_{t+1}$ (remain present) with probability p and becomes absent with probability $1 - p$;
 - if $e \notin E_t$, then $e \in E_{t+1}$ (becomes present) with probability $1 - q$ and remain absent of E_{t+1} with probability q .

The maximum number of edges that may be contained at a given step t is $\binom{n}{2}$. The set of edges is evolving through time and computing E-DynamicScore gives an information about its dynamics. In the following sub sections we establish a solid foundation for understanding its dynamics and its relationship with probabilities p and q .

Note: in the following, the number of edges in a generated graph at step t will be referred to as m_t and the graph density will be referred to as \hat{m}_t . Given an undirected graph $G = (V, E)$, such that $|V| = n$ and $|E| = m$, graph density is $\hat{m} = \frac{2m}{n(n-1)}$

3.2 Known Properties of EMGG

In order to ease the understanding of the dynamics of EMGG instances, some results about EMGG are presented.

First note that the state of each edge is independent of the state of the other edges of the graph, thus, studying the probability of presence/absence of each edge independently from the others is correct. As presented in [5] the transition matrix P for a single edge satisfies:

$$P = \begin{pmatrix} p & 1 - p \\ 1 - q & q \end{pmatrix} \tag{1}$$

The analysis of Markovian processes and more especially the study of two-states markovian processes has shown that for each single edge, the distribution of presence, in the context of EMGG, converges toward a stationary distribution π as long as $|p + q - 1| \neq 1$. The situation $|p + q - 1| = 1$ is discussed after the proof of the theorem. As a stationary distribution of a Markov chain, π satisfies $\pi = \pi P$. The value of vector π is stated in the following theorem:

Theorem 1. *Stationary distribution:*

For p, q probabilities such that $|p + q - 1| \neq 1$, the stationary distribution π is:

$$\left(\frac{1 - p}{2 - p - q} \quad \frac{1 - q}{2 - p - q} \right)$$

Proof: It is sufficient to notice that $\pi = \left(\frac{1-p}{2-p-q} \quad \frac{1-q}{2-p-q} \right)$ is a distribution and that $\pi = \pi P$. □

Thus, the presence of an edge has a Bernoulli distribution of parameter $\pi^* = \frac{1-p}{2-p-q}$ as a stationary distribution. As every edge is independent one from the other, the number of edges has a binomial distribution of parameter $\binom{n}{2}$ and π^* . The situation for which $|p + q - 1| = 1$ as two subcases, either $p = q = 1$ or $p = q = 0$. On the one hand if $p = q = 1$, then graphs produced by EMGG remains unchanged forever. This means $G_t = G_0$ for all t . On the other hand if $p = q = 0$, then produced graphs are 2-periodic and more precisely, E_{t+1} is the complementary of E_t for all t . Thus, in both cases the density of a produced graph does not converge to a stationary distribution.

3.3 EMGG and E-DynamicScore

This subsection presents several key results concerning the Edge-Markovian Graphs Generator (EMGG) and the E-DynamicScore of the graphs it generates. Results stated here are specific cases of the analysis made in the previous section. Every result mentioned in this section will be connected to ones stated above. Firstly, we provide the computation of the density of these graphs, a fundamental quantity in the context of EMGG. The expectation of this quantity is then stated, offering insights into its average behavior. Moving forward, we examine the average DynamicScore across all possible density values. By analyzing this metric, we gain a comprehensive understanding of the dynamics of the EMGG and its relationship with the density parameter. Specifically, we explore the DynamicScore at the fixed point density, uncovering the crucial role played by the probabilities p and q , and elucidating the characteristics that can be derived from this special value. Notably, we establish a meaningful connection between this particular value of the DynamicScore and the values obtained through experimental observations. Through these results, we deepen our understanding of the EMGG and its association with E-DynamicScore, providing valuable insights into the dynamics and quantitative assessment of this stochastic graph generation process.

Density Evolution of Edge-Markovian Graphs

In order to better understand the relationships between EMGG dynamics and the E-DynamicScore metric, we first show that the number of edges is on average close to a quantity depending only on p and q . To that end, we prove the following lemma on the evolution of the density:

Lemma 1. *Evolution of the Density*

Let consider EMGG parameterized by (n, p, q) . Let (G_0, \dots, G_t) be a sequence of graphs produced by EMGG. Then, the expected normalized density for the graph G_{t+1} satisfies the following equation:

$$\hat{m}_{t+1} \simeq f_{p,q}(\hat{m}_t) = \hat{m}_t p + (1 - q)(1 - \hat{m}_t) = (p + q - 1)\hat{m}_t \quad (2)$$

Proof: As the process is a Markov chain, \hat{m}_{t+1} depends only on \hat{m}_t . Second, it is worth mentioning that every edge is independent from the others. The expected number of edges that remain present is $p\hat{m}_t$ while the expected number of edges changing their state from absent to present is $(1 - q)(1 - \hat{m}_t)$. The expected number of edges present at step $t + 1$ is thus the sum of these two quantities. \square

This lemma provides a valuable interpretation of the density expectation, which allows us to further investigate the existence of a fixed density. By analyzing the expectation, we can identify a specific value that represents a fixed point within the computation process. In the context of the function $f_{p,q}$, a fixed point refers to a value m^* for which $f(m^*) = m^*$ holds true. The computation of this fixed point value is carried out according to the procedure outlined in the subsequent lemma.

Lemma 2. *Expected Number of Edges:*

Let G be a graph produced by EMGG(n, p, q) Let \hat{m}_t be the density of graph at step t . Then, as long as $|p + q - 1| < 1$ an expectation value for \hat{m}_t , referred to as m^* , satisfies $f_{p,q}(m^*) = m^*$:

$$m^* = \frac{1 - q}{2 - p - q} \quad (3)$$

Proof: This result comes from finding a fixed point to the function $f_{p,q}$ \square

This fixed point value matches with the probability of presence of an active edge in the stationary regime. It is not surprising as it gives, in both case, the average and expected value of the graph density. These findings enable us to gain deeper insights into the dynamics of the system and the properties associated with the EMGG, paving the way for a more comprehensive understanding of its behavior.

3.4 Relationship with the DynamicScore

This section explores the relationship between the Edge-Markovian Graphs Generator (EMGG) and the DynamicScore, focusing on the computation of an expectation value regardless of the graph's density. The following theorem provides a precise value of this expectation, elucidating the crucial role played by the parameters p and q in this context:

Theorem 2. *Average General DynamicScore*

Let G be a graph produced by $EMG(n, p, q)$. Let \hat{m}_t be the density of graph at step t . Then, in average:

$$\mathcal{D}_t^E = 1 - \frac{p\hat{m}_t}{1 + q(\hat{m}_t - 1)} \tag{4}$$

Proof: The proof consists in finding the average number of edges in $E_t \Delta E_{t+1}$ and in $E_t \cup E_{t+1}$. For the first one, it consists in computing, on average, the number of edges which state is changing. Assuming the density of edges at t is \hat{m}_t , then the density of edges that change from present to absent is on average $(1 - p)\hat{m}_t$ and the density of newly present edges is on average $(1 - q)(1 - \hat{m}_t)$. Therefore, the size of $E_t \Delta E_{t+1}$ is on average:

$$|E_t \Delta E_{t+1}| = (1 - p)\hat{m}_t + (1 - q)(1 - \hat{m}_t)$$

For computing the union size, it is sufficient to notice that it contains all the present edges at step t plus appearing edges $(1 - q)(1 - \hat{m}_t)$. Thus, the size of the union is in average:

$$|E_t \cup E_{t+1}| = \hat{m}_t + (1 - q)(1 - \hat{m}_t)$$

It is therefore possible to estimate the average DynamicScore:

$$\mathcal{D}_t^E = \frac{(1 - p)\hat{m}_t + (1 - q)(1 - \hat{m}_t)}{\hat{m}_t + (1 - q)(1 - \hat{m}_t)} = 1 - \frac{p\hat{m}_t}{1 + q(\hat{m}_t - 1)}$$

□

This result must be evaluated for densities close to m^* . Indeed, the distribution of the edges follows a binomial law of parameters $\binom{n}{2}$ and π^* . Therefore most values of $|E_t|$ taken by generated graphs are close to the expected value of the binomial law: $\binom{n}{2}\pi^*$. This implies density of these graphs are close to $\pi^* = m^*$. Combining this theorem with the fixed point density of generated graphs provides DynamicScore at the fixed point density:

Theorem 3. *E - DynamicScore in Average around m^* :*

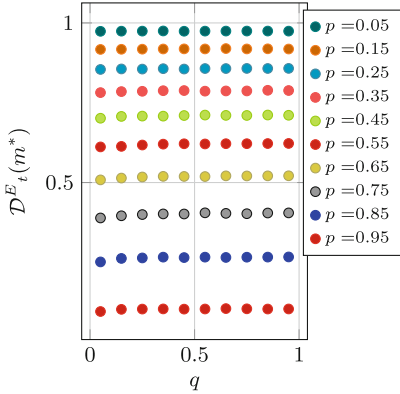
For all p, q such that $|p + q - 1| < 1$

$$\mathcal{D}_t^E(m^*) = 2 \frac{1 - p}{2 - p}$$

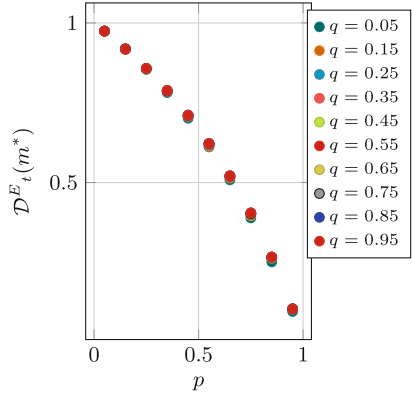
Moreover, $\mathcal{D}_t^E(m^*)$ may take all the values from 0 to 1.

Proof: It results from the combination of both Theorem 2 and Lemma 2. □

Notably, the average DynamicScore is independent of the value of q , and it exhibits a decreasing trend as p increases. The range of possible values for the DynamicScore ranges from 0 to 1, indicating its ability to capture the extent of changes in the graph. To illustrate these findings, several figures are presented.



(a) E-DynamicScore Average vs. q .



(b) E-DynamicScore Average vs. p .

Fig. 3. Average dynamic score as a function of the parameters p and q . On the left, the parameter p is set and the parameter q ranges from 0.05 to 0.95. One may notice that for a fixed value of parameter p , the average dynamicScore does not depend on q . On the right, the parameter q is set and the parameter p ranges from 0.05 to 0.95. As observed with the picture on the left, the average dynamicScore does not depend on q so all the marks are mingled.

These figures have been obtained through simulations, considering various values of p and q , both ranging from 0 to 1, while excluding the endpoints. These visual representations offer a good understanding of the relationship between EMGG instances, their DynamicScore on average, and the parameters p and q . By examining these figures, we obtain experimental confirmation and deeper understanding of the behavior and characteristics of the EMGG, corroborating the insights provided by the above-stated theorem, particularly in relation to the DynamicScore. The impact of the parameter q on the average DynamicScore is found to be negligible, whereas parameter p appears to be more influential in determining its value. Notably, it is observed that the DynamicScore can encompass the entire range from 0 to 1 as p varies from 1 to 0 (Fig. 3).

4 Conclusion and Open Problems

In this work, a new metric called DynamicScore has been presented. This metric, proposed for both edges and vertices, quantifies the evolution of the dynamics of dynamic graphs. It has been demonstrated that the Preferential Attachment growing model generates graphs with dynamics that tends toward zero. This implies that after numerous iterations, the dynamic graph undergoes minimal changes, resulting in a stability of the properties within the generated graphs. However, the dynamics of real complex networks is not solely reliant on the creation of vertices and edges but also on the removal of vertices and edges. This leads us to the following open question: (i) *given a specific dynamic graphs gen-*

erator, is there a relationship between *DynamicScore* values and the preservation of properties in dynamic graphs?

The second studied generator was the Edge-Markovian Graphs Generator. The mechanics of this generator is based on two probabilistic parameters, p and q , driving the states of edges that can be present or absent. After an in-depth analysis of the average density of the generated graphs, using *DynamicScore*, it has been shown that the dynamics of such graphs is only driven by probability parameter p . The analysis relies on the Markovian nature of the generator, which prompts two additional open questions: (ii) *if the evolution/generative process exhibits Markovian characteristics in the evolution of edges, does the value of E-DynamicScore remain nearly constant?* and (iii) *conversely, if the value of DynamicScore remains constant, does this indicate that the evolution process is Markovian?*

References

1. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999)
2. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.: Complex networks: structure and dynamics. *Phys. Rep.* **424**(4–5), 175–308 (2006)
3. Bridonneau, V., Guinand, F., Pigné, Y.: Dynamic Graphs Generators Analysis?: An Illustrative Case Study. LITIS, Le Havre Normandie University, Tech. rep. (2022)
4. Casteigts, A., Flocchini, P., Quattrociocchi, W., Santoro, N.: Time-varying graphs and dynamic networks. In: Frey, H., Li, X., Ruehrup, S. (eds.) *Ad-hoc, Mobile, and Wireless Networks*, pp. 346–359. Springer, Berlin Heidelberg (2011). https://doi.org/10.1007/978-3-642-22450-8_27
5. Clementi, A.E.F., Macci, C., Monti, A., Pasquale, F., Silvestri, R.: Flooding time of edge-Markovian evolving graphs. *SIAM J. Discret. Math.* **24**(4), 1694–1712 (2010)
6. Ferreira, A., Viennot, L.: A note on models, algorithms, and data structures for dynamic communication networks. Report, INRIA (2002). <https://hal.inria.fr/inria-00072185>
7. Harary, F., Gupta, G.: Dynamic graph models. *Math. Comput. Model.* **25**(7), 79–87 (1997)
8. Holme, P., Saramäki, J.: Temporal networks. *Phys. Rep.* **519**(3), 97–125 (2012). *Temporal Networks*
9. Kostakos, V.: Temporal graphs. *Phys. A* **388**(6), 1007–1023 (2009)
10. Xuan, B.B., Ferreira, A., Jarry, A.: Computing shortest, fastest, and foremost journeys in dynamic networks. *Int. J. Found. Comput. Sci.* **14**(02), 267–285 (2003)



A Novel Method for Vertex Clustering in Dynamic Networks

Devavrat Vivek Dabke¹(✉) and Olga Dorabiala²

¹ Princeton University, Princeton, USA
ddabke@princeton.edu

² University of Washington, Washington, USA
olgad400@uw.edu

Abstract. In this paper, we introduce *spatiotemporal graph k -means* (STG k M), a novel, unsupervised method to cluster vertices within a dynamic network. Drawing inspiration from traditional k -means, STG k M finds both short-term dynamic clusters and a “long-lived” partitioning of vertices within a network whose topology is evolving over time. We provide an exposition of the algorithm, illuminate its operation on synthetic data, and apply it to detect political parties from a dynamic network of voting data in the United States House of Representatives. One of the main advantages of STG k M is that it has only one required parameter, namely k ; we therefore include an analysis of the range of this parameter and guidance on selecting its optimal value. We also give certain theoretical guarantees about the correctness of our algorithm.

Keywords: Vertex clustering · Dynamic networks · Graph clustering · Community detection · k -means

1 Introduction

Dynamic graphs are becoming increasingly prevalent mathematical structures as we collect more detailed data on the world around us. Though graphs have traditionally been studied as static objects, the dynamic setting better captures systems that evolve over time. Also called “time-varying” or spatiotemporal graphs, they extend static graphs by permitting edges to change over time, and they inherently reflect many systems, e.g., road networks, online communities, and epidemic spread. Since they are much less understood than their static counterparts, they pose an exciting and rich area of study.

Much of the literature on dynamic graphs focuses on extending well-known concepts from the static case like connectivity [15], optimal routing [6], induced dynamical systems [8], and more. Our work fits into this foundational literature by extending the notion of vertex clustering for the purpose of community

Both authors contributed equally to this work.

See <https://github.com/dynestic/stgkm> for the associated code for this project.

We would like to acknowledge J. Nathan Kutz (U. of Washington) for his support.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
H. Cherifi et al. (Eds.): COMPLEX NETWORKS 2023, SCI 1142, pp. 445–456, 2024.
https://doi.org/10.1007/978-3-031-53499-7_36

detection. Graph clustering is a fundamental tool for network analysis, with applications across the social and natural sciences, and we seek to bring this tool to the dynamic setting. In dynamic graph clustering, we find a partition of graph vertices that takes into account both spatial similarity—so that there are many edges within a cluster and relatively few between clusters—and temporal similarity, so clusters stay consistent over time. These partitions help us detect latent community structures.

In this paper, we propose a method we call *spatiotemporal graph k -means* (STG k M) that is able to track the multi-scale relationships between graph vertices. STG k M applies a two-phase clustering approach, wherein the first phase outputs an assignment for each vertex at every time step and the second phase produces a single, long-term partition of vertices based on historical cluster membership. STG k M identifies communities of interest and automatically tracks their evolution over time. To validate our method, we provide certain theoretical guarantees and showcase the utility of STG k M on synthetic and real-world datasets.

2 Related Work

In static graphs, vertex clustering has a broad literature with interdisciplinary interest and there has been a push to extend these results to the dynamic setting. Most approaches to dynamic community detection find clusters independently at each time step and then use aggregation to successively infer relationships between partitions [11]. These methods are often unable to achieve temporal smoothness and inevitably do not capture the dynamics of the network [12]. Another subset of methods first constructs a single coupling graph that summarizes the temporal properties of the dynamic network and then runs a classic community detection method on this graph [22]. As with aggregation, the use of coupling graphs results in a loss of temporal information.

Evolutionary clustering addresses this shortcoming through a unified framework, where clusters are iteratively formed based on current network structure and previous partitions. A cost function regulates the tradeoff between cluster quality at each snapshot and cluster consistency [4]. This framework has been successfully adopted and refined [5, 19]. Other lines of research extend static community detection using online algorithms [23], machine learning [24], or systems-based approximation algorithms [9]. These papers leverage diverse methods to contend with the sometimes staggering size of dynamic networks.

Our method, STG k M, develops a unified framework akin to, but distinct from, evolutionary clustering [4]. STG k M, achieves temporal smoothness by restricting the search space of new cluster centers based on temporal reachability from previous centers. In addition, our method goes further than existing techniques to also extract long-lived communities of vertices based on historical dynamic cluster membership. STG k M is the graph analogue to our previously developed point-based method [10]. We first introduced a notion of STG k M in an extended abstract with some preliminary evidence of its effectiveness, but we have since refined our approach and this paper provides our complete results [7].

Finally, we note that there are numerous methods to group vertices within a dynamic network, each with its own motivation, challenges, and rich literature. Our method of vertex partitioning prioritizes long-term stable connections and our main theoretical result highlights the relationship to the distinct concept of connected components. However, there are many other interesting notions of connectivity [3] with variants in stochastic settings [1] along with other related problems, like motif detection [13], centrality measurement [2, 14, 18, 25], and even novel frameworks for capturing properties of dynamic networks [17].

3 Spatiotemporal Graph k -means (STG k M)

Our goal is to partition a vertex set given a dynamic graph. In STG k M, we construct a partition by finding *central* nodes to represent each cluster and then assigning each remaining vertex based on its closest central node.¹ Just as with k -means, we define the problem of finding good clusters as a minimization problem; our novel objective has a unified formulation over space and time that predicts a partition for each vertex at every time step. After pre-processing, STG k M consists of two phases: in a single pass of Phase 1, the algorithm outputs vertex membership and dynamic cluster center journeys; in Phase 2, we extract the long-lived communities from the graph.

Setup

As input data, we need: a (finite) vertex set V , a (finite) time set $\mathbb{T} \subset \mathbb{N}$, a dynamic graph $\mathcal{G} = (V, E^t)_{t \in \mathbb{T}}$ where $E^t \in V \times V$, and an optional non-negative cost function $\omega^t : V \times V \rightarrow \mathbb{R}$ for all $t \in \mathbb{T}$. Our parameters are $k \in \mathbb{N}$, the number of clusters, and—optionally— $\lambda \in \mathbb{N}$, the maximum cluster center drift, and $\gamma \in \mathbb{Z}_{\geq 0}$, the drift time window.

Pre-processing

For all pairs of vertices across time, we compute and store the s -journey δ , see [15] for details. The value of $\delta^t(u, v)$ is the length of the shortest journey (i.e. dynamic path) starting at vertex u at time t and ending at vertex v . If no such journey exists, it assigns $+\infty$. Though not a true metric (it is missing symmetry and coincidence), this function has the same purpose as a distance in classical k -means.² We also define the related true metric $\tilde{\delta}^t(u, v) \triangleq \delta^t(u, v) + \delta^t(v, u)$ with the additional convention that $\tilde{\delta}^t(u, u) = 0$.

Phase 1

Given a fixed value of k , the first phase of STG k M selects a set of k vertices to serve as cluster centers and assigns each vertex to a cluster at every time step.

¹ Our approach is perhaps more analogous to k -medoids, but in a network context, the distinction between k -means and k -medoids is not obvious.

² If no weight functions are provided or if the weight functions only output natural numbers, then δ will assign only natural numbers.

Vertices have the flexibility to switch cluster membership at every time step, but cluster centers are constrained by drift parameters λ and γ .

Natural Objective. The natural extension of k -means would be to optimize the objective function in Expression 1:

$$\min_{c \in \mathcal{C}, W \in \mathcal{W}} \sum_{t \in \mathbb{T}} \sum_{u \in V} \sum_{j \in [k]} W_{u,j}^t \cdot \tilde{\delta}^t(u, c_j^t) \tag{1}$$

where we minimize over cluster centers \mathcal{C} and assignment tensors \mathcal{W} . Formally, \mathcal{C} is the set of all sequences of length $|\mathbb{T}|$ where each element is an ordered subset of V with k elements; $\mathcal{W} \triangleq \{0, 1\}^{|\mathbb{T}| \times |V| \times k}$ such that $\sum_{j \in [k]} W_{u,j}^t \geq 1$. Note that we allow vertices to belong to multiple clusters simultaneously, and each vertex is assigned to at least one cluster at every time step.

Objective with Regularization. Optimizing Expression 1 is NP-hard³, so we instead iteratively optimize a modified objective function that restricts the search space. We begin by choosing initial cluster centers c^0 to be the nodes that are most closely connected to all others at t_0 . When there are ties, we sample randomly. OlgAt each time t henceforth, we assume that we have chosen optimal cluster centers c^s for all $s < t$, and we minimize Expression 2.

$$\min_{c, W} \sum_{u \in V} \sum_{j \in [k]} W_{u,j}^t \cdot \delta^t(u, c_j^t) \tag{2}$$

such that $\delta^{t-q}(c_j^{t-1}, c_j^t) \leq \lambda$, where $1 \leq q \leq \gamma$ and $1 \leq j \leq k$

The constraint in Expression 2 imposes that the center of a given cluster can only switch from vertex u to vertex v if the distance between them is no more than λ for the previous γ time steps. This regularization serves two purposes: first, it associates dynamic clusters between time steps; second, it restricts the search space for cluster centers⁴. As we decrease λ or increase γ , we decrease the number of potential centers at time t and enforce stricter cluster consistency; see Fig. 1 for an example.

In practice, we update the center of a cluster only if the objective is improved. When we encounter the case where selecting new cluster centers is infeasible, we update clusters individually instead of jointly. Our algorithm terminates until either the clusters stabilize or we reach a maximum number of iterations.

³ To see why, observe that k -medoids is NP-hard [20].

⁴ In the worst case, e.g. when the graph is complete at every time step, optimizing this objective is still NP-hard, but in practice, it makes STG k M tractable.

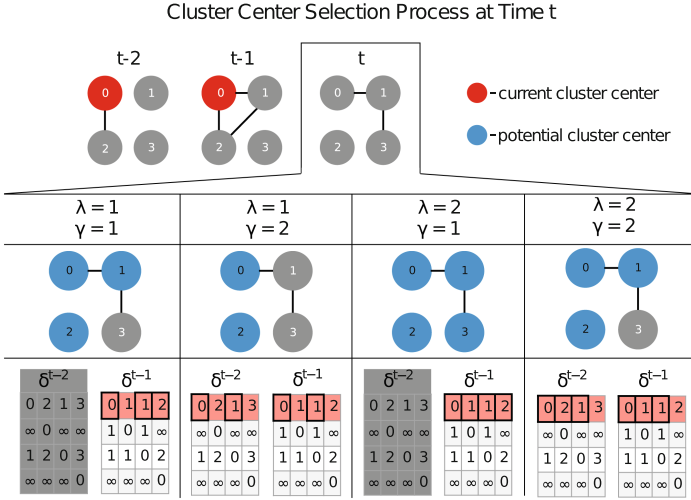


Fig. 1. At time t , c_0^t is chosen based on c_0^{t-1} . The drift time window γ determines for how many previous time steps centers must be within maximum drift λ of one another. The objective in Expression 2 is evaluated for all potential cluster centers; the center that minimizes the objective is chosen.

Phase 2

By building on Phase 1, Phase 2 of STGkM aims to identify the long-lived partitions of graph vertices. The output is an assignment of communities containing vertices with the most similar spatiotemporal characteristics. Intuitively, we expect vertices with similar partitioning histories to belong to the same persisting community in the long run.

Recall that the *Hamming distance* is defined by counting the number of entries where two matrices disagree: $H(u, v) \triangleq |\{(t, k) : W_{u,k}^t \neq W_{v,k}^t\}|$. Using this distance, we define similarity $\text{sim}(u, v)$ as

$$\text{sim}(u, v) \triangleq 1 - \frac{H(u, v)}{|\mathbb{T}|} \tag{3}$$

This definition gives us a powerful way to compare all pairs of vertices. Since $\text{sim}(\cdot, \cdot)$ is compatible with traditional clustering techniques, we input it to agglomerative clustering. We then output the resulting partition to get a clustering of the vertices based on long-lived communities as desired.

4 Results

4.1 Algorithmic Analysis

The main feasibility issue with STGkM arises from finding new cluster centers at every time step. Evaluating all possible subsets of size k of $|V|$ is NP-hard.

The objective in Expression 2 does not obviate this possibility in the worst case, even though in practice, the added regularization results in a sufficiently fast algorithm. There are other strategies we could deploy to minimize our objective (e.g. greedy algorithms, subsampling, further constraints on cluster center drift) that may be theoretically efficient, but we did not explore these possibilities in detail, since our chosen strategy works in practice. Also, we do not provide formal guarantees on approximation quality for Expression 2.

4.2 Connected Components

Although the clusters that STGkM generates are distinct from connected components, we can find connected components under certain conditions, as presented in Theorem 1. Though our method may not be most efficient way to find connected components (and there are other notions of connected components), our theoretical result provides evidence that STGkM can find interesting partitions.

Definition 1 (Dynamic Connected Component). *Vertices u, v are (dynamically) connected if there exists a finite journey from u to v and from v to u over all time steps. A set of vertices U (where $U \subseteq V$) is a (dynamic) connected component if all vertices in this set are connected and there is no vertex in $V \setminus U$ that is connected to a vertex in U .*

Lemma 1. *For two vertices u, v in distinct connected components, there exists some time step t_0 such that $\tilde{\delta}^{t_0}(u, v) = \infty$.*

Proof. By definition, if u, v are not connected, then there must be some time step t_0 at which there is no finite journey from u to v or v to u . \square

Definition 2 (Self-Connected). *A dynamic network is self-connected if each vertex is connected to itself.*

Lemma 2. *In a self-connected dynamic network, connectivity is an equivalence relation and the connected components are the respective equivalence classes.*

Proof. By construction, connectivity is symmetric. By self-connection, connectivity is reflexive. Connectivity is also transitive: if vertex u is connected to v and v is connected to w , then there exists a journey from u to w via v for all time steps. By construction, each connected component only contains vertices that are connected and no other vertices, so it is an equivalence class. \square

We now make use of two further concepts: a *non-stranding* dynamic graph is one where for every time step, every vertex has at least one edge, and *holding*, where each vertex has a self-loop at every time step [15]. We introduce Lemma 3, which is related to (but distinct from) Proposition 3.5 in [15].

Lemma 3. *For two vertices u, v in distinct connected components in a non-stranding, holding dynamic network, there exists some time step t_0 such that $\tilde{\delta}^t(u, v) = \infty \forall t > t_0$.*

Proof. We proceed by induction. For our base case, by Lemma 1, there exists t_0 such that $\tilde{\delta}^{t_0}(u, v) = \infty$. Therefore, either $\delta^{t_0}(u, v) = \infty$ or $\delta^{t_0}(v, u) = \infty$, so (without loss of generality) we assume $\delta^{t_0}(u, v) = \infty$.

For our inductive step, we will show that if for some t that $\delta^t(u, v) = \infty$, then $\delta^{t+1}(u, v) = \infty$. By holding $\delta^t(u, u) = 1$ and so $\delta^t(u, v) \leq 1 + \delta^{t+1}(u, v)$, which immediately implies that $\delta^{t+1}(u, v) = \infty$. By induction, for all $t > t_0$, $\delta^t(u, v) = \infty$ and so $\tilde{\delta}^t(u, v) = \infty$ as desired. \square

Theorem 1. *Given a holding, non-stranding dynamic graph with k connected components, the partition of vertices induced by the optimal solution to Expression 1 is exactly the connected components given sufficient time.*

Proof. For every pair of vertices u, v that are not connected, there exists some time step such that $\tilde{\delta}^t(u, v) = \infty$ for all $t > t_{u,v}$ by Lemma 3. For a pair of vertices u, v , denote $t_{u,v}$ to be the minimum such time step if they are disconnected or 0 otherwise. Let $t^* = \max\{t_{u,v}\}$. This notion is well-defined because every vertex is in a connected component at least with itself by Lemma 2.

For all t such that $t \geq t^*$, note that $\tilde{\delta}^t(u, v) = \infty$ if and only if u, v are in different connected components. If two entries in c^t are in the same connected component, then there must be one vertex v that is not connected to any cluster center in c^t by the pigeonhole principle and thus $\tilde{\delta}^t(v, c_j^t) = \infty$ for all j . At least one entry in $W_{v,\bullet}^t$ must be 1 by construction and thus

$$\sum_{j \in [k]} W_{v,j}^t \cdot \tilde{\delta}^t(u, c_j^t) = \infty$$

Conversely, we can select vertices c_j with $j \in [k]$ such that each is in a distinct connected component. Now, construct W^t such that $W_{u,j}^t$ is 1 when u and c_j are connected and is 0 otherwise. With this construction, the sum below is finite:

$$\sum_{u \in V} \sum_{j \in [k]} W_{u,j}^t \cdot \tilde{\delta}^t(u, c_j^t)$$

and this constructed W^t is optimal. For two connected vertices u, v , $W_{u,\bullet}^t = W_{v,\bullet}^t$ so $\text{sim}(u, v) \geq 1 - \epsilon$ where $\epsilon = \frac{t^* \times k}{|\mathbb{T}|}$. For two disconnected vertices x, w , there exists at least one index j such that $W_{x,j}^t \neq W_{w,j}^t$ so $\text{sim}(x, w) < \epsilon + \frac{k-1}{k}$. With sufficiently large $|\mathbb{T}|$, we will correctly separate these clusters. \square

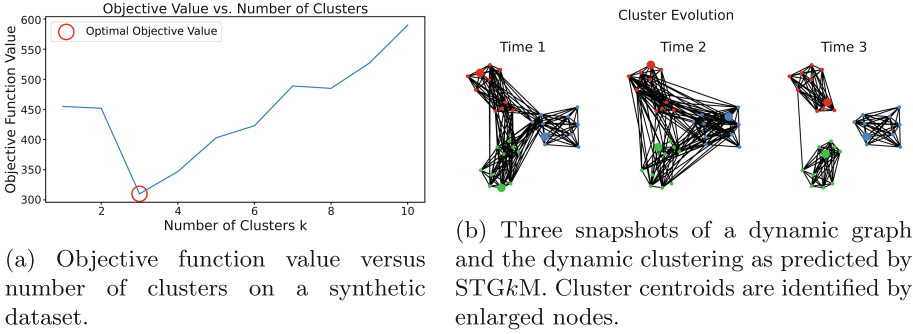


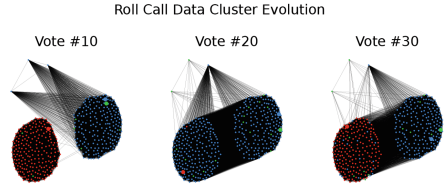
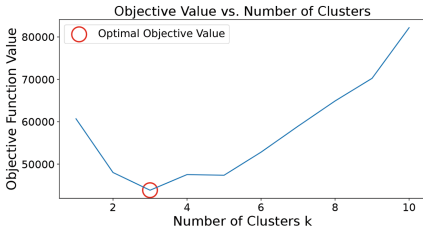
Fig. 2. STGkM on a synthetic dataset, consisting of three ground-truth clusters.

4.3 Experimental Insights

Choosing k . Perhaps the greatest challenge in using a k -means-based approach for clustering is determining the optimal number of clusters k . With regard to classical k -means, one of the most common methods for choosing k is the Elbow Method [16], wherein the sum of square error of each cluster is calculated, and the value of k which results in the most extreme difference (the elbow) is chosen. We employ an approach similar to the Elbow Method for STGkM.

We calculate the value of Expression 2 for every value of k being considered, and seek the value of k for which the objective is minimized. It is important to note, that as opposed to the Elbow Method, which finds the most extreme difference in objective values, we seek the minimal value. The reason for this is that we allow for vertices to belong to multiple clusters simultaneously. Unlike classical k -means where increasing k results in points getting progressively closer to their centers, in STGkM, increasing k is likely to cause vertices to be assigned to progressively more clusters simultaneously. Consider the case of n vertices in a clique: cluster centers will be assigned only to their own cluster, whereas every other vertex will be assigned to all clusters. The value of Expression 2 is thus $(n - k)k$. Restricting to small k , the objective value is minimized when $k = 1$ and increases as k increases. In a clique, we would expect $k = 1$.

Synthetic Data. We begin by applying STGkM to a synthetic dataset, consisting of three clusters with 10 fully connected nodes in each cluster, tracked over 20 time steps. The result is a dynamic graph with 30 nodes and 300 edges at each time t . At every time step we randomly choose up to 30 edges to remove within clusters and up to 30 edges to add between clusters. We run STGkM with $\lambda = 1, \gamma = 1$. Following the method for choosing k described previously, we set $k = 3$. The selection process for choosing k is shown in Fig. 2a, and a snapshot of the evolution of detected clusters is shown in Fig. 2b. As expected, three communities persist throughout the duration of the simulation.



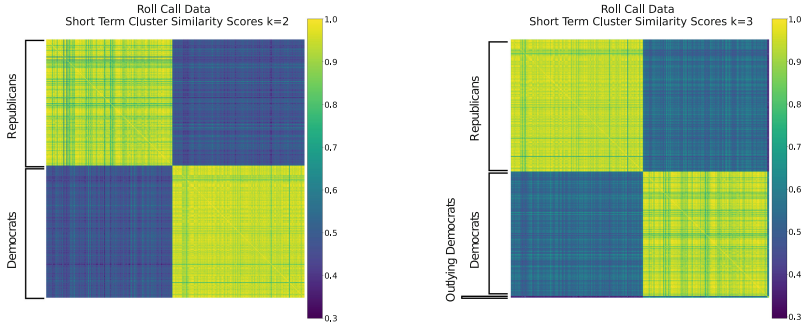
(a) Objective function value versus number of clusters on the Roll Call dataset.

(b) Snapshots of the dynamic graph created by the Roll Call dataset. Cluster centroids are identified by enlarged nodes.

Fig. 3. STGkM on the Roll Call dataset.

Detecting Political Parties. To demonstrate the utility of STGkM on a larger, real-world dataset, we turn to politics. Communities naturally arise in politics, particularly in recent years where we have witnessed polarization with political figures consistently voting along party lines. Taking inspiration from [21], we form a dynamic graph based on 100 roll call votes from the House of Representatives between June 21, 2023 and July 27, 2023. Each vote is a time step, each representative is a node, and nodes are connected if they vote the same way on a bill. Possible votes are “Yea”, “Nay”, and “Present”. If a representative does not cast a vote, they have no connecting edges for that vote. The ground truth communities are representative’s affiliated political parties. By running STGkM on the roll call graph, we can identify the communities of representatives that vote similarly and observe how those communities evolve over time.

We choose our maximum center drift to be $\lambda = 1$ and our time connectivity to be $\gamma = 5$. Intuitively, we expect $k = 2$, but our k selection process recommends $k = 3$, as seen in Fig. 3a. We find that when we run STGkM with $k = 2$, we correctly separate Democrats and Republicans in our long term clusters, but interestingly enough, when we run STGkM with $k = 3$, we additionally find a sub-community of three Democrats. Upon further investigation, we find that these three Democrats, Rep. Joaquin Castro, Rep. Emanuel Cleaver, and Rep. Michael Kelly, very often vote “Present” together, as opposed to the majority Democratic party vote. The evolution of our dynamic clusters is visualized in Fig. 3b. We observe two large persistent clusters in red and blue. We also see how in some votes, such as #20 where most representatives vote identically, almost all nodes are assigned to one cluster. The third, green cluster often contains the three outlying Democrats, as well as other “Present” voters over various votes. Figures 4a and 4b visualize the similarity scores, as defined in Eq. 3, between the cluster assignment histories for each pair of representatives. The rows and columns of the similarity matrix are ordered according to the long-term communities discovered by STGkM. In Fig. 4a, these clusters correspond to Republicans followed by Democrats, while in Fig. 4b, the three outlying Democrats are moved



(a) Short term clustering similarity between nodes in the Roll Call dataset using STG k M with $\gamma = 5$, $\lambda = 1$, and $k = 2$.

(b) Short term clustering similarity between nodes in the Roll Call dataset using STG k M with $\gamma = 5$, $\lambda = 1$, and $k = 3$.

Fig. 4. Similarity matrices of the short term clustering similarity between nodes in the Roll Call dataset. Rows and columns of the matrices are organized by detected long-term community membership.

to the final three rows and columns of the matrix. We observe a distinct color difference between these three rows and the remainder of the matrix, demonstrating that the similarity between the outlying Democrats and remaining Democrats is much lower.

5 Conclusion

We introduce spatiotemporal graph k -means (STG k M) for community detection by vertex clustering on dynamic graphs. This approach is unified over space and time and gives us the ability to analyze both the short- and long-term partitions of graph vertices, monitor the multi-scale relationships between communities, and has just three explainable parameters, only one of which is required. We provide a principled approach to estimating the required parameter: the number of clusters k . We also state some theoretical guarantees that explain clustering behavior under certain conditions. Finally, we carry out experiments on both a synthetic and real world dataset to empirically validate STG k M.

In our future work, we seek to improve the efficiency of STG k M, both in practice and in theory. As STG k M is applied to larger datasets, further approximation strategies will be necessary to ensure feasibility. We would like to provide guidance on the quality and convergence of our approximation strategies. The theoretical guarantees in this paper are only correct under narrow conditions, so we would like to provide more contexts in which clustering is assured to work correctly. We will also explore online extensions of STG k M, where we explore dynamic graphs in real-time. Finally, we seek a characterization of the expected properties of STG k M in a stochastic setting with the presence of noise.

References

1. Becker, R., et al.: Giant components in random temporal graphs. arXiv preprint [arXiv:2205.14888](https://arxiv.org/abs/2205.14888) (2022)
2. Bergamini, E., Meyerhenke, H.: Approximating betweenness centrality in fully dynamic networks. *Internet Math.* **12**(5), 281–314 (2016)
3. Casteigts, A., Flocchini, P., Quattrociochi, W., Santoro, N.: Time-varying graphs and dynamic networks. *Int. J. Parallel Emergent Distrib. Syst.* **27**(5), 387–408 (2012)
4. Chakrabarti, D., Kumar, R., Tomkins, A.: Evolutionary clustering. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 554–560 (2006)
5. Chi, Y., Song, X., Zhou, D., Hino, K., Tseng, B.L.: Evolutionary spectral clustering by incorporating temporal smoothness. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 153–162 (2007)
6. Cleveland, J., et al.: Introducing tropical geometric approaches to delay tolerant networking optimization. In: 2022 IEEE Aerospace Conference (AERO), pp. 1–11 (2022)
7. Dabke, D.V., Dorabiala, O.: Spatiotemporal graph k-means. In: Proceedings of the Communities in Networks ComNets @ NetSci 2023 (2023)
8. Dabke, D.V., Karntikoon, K., Aluru, C., Singh, M., Chazelle, B.: Network-augmented compartmental models to track asymptomatic disease spread. *Bioinform. Adv.* **3**, vbad082 (2023)
9. DiTursi, D.J., Ghosh, G., Bogdanov, P.: Local community detection in dynamic networks. In: 2017 IEEE International Conference on Data Mining (ICDM), pp. 847–852 (2017)
10. Dorabiala, O., Webster, J., Kutz, N., Aravkin, A.: Spatiotemporal k-means. arXiv preprint [arXiv:2211.05337](https://arxiv.org/abs/2211.05337) (2022)
11. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3–5), 75–174 (2010)
12. Görke, R., Maillard, P., Schumm, A., Staudt, C., Wagner, D.: Dynamic graph clustering combining modularity and smoothness. *J. Exp. Algorithmics* **18**, 1–1 (2013)
13. Gurukar, S., Ranu, S., Ravindran, B.: Commit: a scalable approach to mining communication motifs from dynamic networks. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, SIGMOD 2015, pp. 475–489. Association for Computing Machinery, New York (2015)
14. Habiba, C.T., Tanya, Y.: Berger-Wolf. Betweenness centrality measure in dynamic networks, Technical Report 19, DIMACS (2007)
15. Hylton, A., et al.: A survey of mathematical structures for lunar networks. In: 2022 IEEE Aerospace Conference (AERO), pp. 1–17 (2022)
16. Kodinariya, T.M., Makwana, P.R., et al.: Review on determining number of cluster in k-means clustering. *Int. J.* **1**(6), 90–95 (2013)
17. Latapy, M., Viard, T., Magnien, C.: Stream graphs and link streams for the modeling of interactions over time. *Soc. Netw. Anal. Min.* **8**(1), 61 (2018)
18. Lerman, K., Ghosh, R., Kang, J.H.: Centrality metric for dynamic networks. In: Proceedings of the Eighth Workshop on Mining and Learning with Graphs, MLG 2010, pp. 70–77. Association for Computing Machinery, New York (2010)
19. Lin, Y.-R., Chi, Y., Zhu, S., Sundaram, H., Tseng, B.L.: FacetNet: a framework for analyzing communities and their evolutions in dynamic networks. In: Proceedings of the 17th International Conference on World Wide Web, pp. 685–694 (2008)

20. Megiddo, N., Supowit, K.J.: On the complexity of some common geometric location problems. *SIAM J. Comput.* **13**(1), 182–196 (1984)
21. Reda, K., Tantipathananandh, C., Johnson, A., Leigh, J., Berger-Wolf, T.: Visualizing the evolution of community structures in dynamic social networks. In: *Computer Graphics Forum*, vol. 30, pp. 1061–1070. Wiley Online Library (2011)
22. Rossetti, G., Cazabet, R.: Community discovery in dynamic networks: a survey. *ACM Comput. Surv.* **51**(2), 1–37 (2018)
23. Ruan, B., Gan, J., Wu, H., Wirth, A.: Dynamic structural clustering on graphs. In: *Proceedings of the 2021 International Conference on Management of Data*, pp. 1491–1503 (2021)
24. Yao, Y., Joe-Wong, C.: Interpretable clustering on dynamic graphs with recurrent graph neural networks. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 4608–4616 (2021)
25. Yen, C.-C., Yeh, M.-Y., Chen, M.-S.: An efficient approach to updating closeness centrality and average path length in dynamic networks. In: *2013 IEEE 13th International Conference on Data Mining*, pp. 867–876 (2013)



A Particle Method for Continuous Hegselmann-Krause Opinion Dynamics

Christoph Börgers¹, Natasa Dragovic²(✉), Anna Haensch^{1,3},
and Arkadz Kirshtein¹

¹ Department of Mathematics, Tufts University, Medford, MA, USA
{christoph.borgers,anna.haensch,arkadz.kirshtein}@tufts.edu

² Department of Mathematics, University of St. Thomas, St. Paul, MN, USA
natasa.dragovic@stthomas.edu

³ Data Intensive Studies Center, Tufts University, Medford, MA, USA

Abstract. We derive a differential-integral equation akin to the Hegselmann-Krause model of opinion dynamics [R. Hegselmann and U. Krause, JASSS, vol. 5, 2002], and propose a particle method for solving the equation. Numerical experiments show second-order weak convergence of the method. We also show that our differential-integral equation can equivalently be stated as a system of differential equations. An integration-by-parts argument that would typically yield an *energy dissipation* inequality in physical problems yields a *concentration* inequality here, showing that a natural measure of concentration increases monotonically.

Keywords: opinion dynamics · Hegselmann-Krause model · bounded confidence model · particle method

1 Introduction

People’s opinions and beliefs are influenced in complex ways by families, friends, colleagues, media, as well as politicians and other mega-influencers [1, 5, 6, 11, 13]. In recent decades, attempts have been made to understand aspects of this process using mathematical modeling and computational simulation; for surveys on the subject of opinion dynamics, see for instance [2, 3, 25, 28, 30, 31].

Many models of opinion dynamics are based on the assumption that we are influenced more easily by people whom we *almost* agree with to begin with than by those whose views starkly differ from ours. A similar but more general phenomenon is known as *biased assimilation* among psychologists—our tendency to filter and interpret information in such a way that it supports our preconceived notions [22]. Models of opinion dynamics based on this assumption are known as *bounded confidence models* [3, 12, 26]. A popular example is due to Hegselmann and Krause [18, 19], building on earlier work by Krause [20, 21]. It has been studied extensively in the literature (see for instance [23, 24], and [7]), and will be our starting point here. The Weissbuch-Deffuant model [34] is very close to that of Hegselmann and Krause; while Hegselmann and Krause assume that

each opinion holder responds to all nearby opinions simultaneously, Weissbuch and Deffuant assume random encounters between pairs of opinion holders with similar views. For other bounded confidence models, see [4, 15, 32].

The original Hegselmann-Krause model is discrete in both time and opinion space. Similar models that are continuous in time [29], opinion space [33], or opinion space and time [16] have been proposed as well. We are particularly interested in fully continuous models, since we plan, in future work, to explore the response of candidates to a dynamic electorate. In a previous paper, we have already discussed the response of candidates to a *static* electorate [8]. We want to describe candidate dynamics in opinion space by ordinary differential equations, and find that easiest to do in clean and natural ways if the opinion dynamics of the electorate are described fully continuously. We note that “continuous” does not mean the same to all authors in this field. For instance, in Lorenz’s earlier papers [23, 24], the dynamics are discrete in both opinion space and time. The word “continuous” appears in the titles of both papers, but it indicates merely that the opinions can take arbitrary real values. To us, by contrast, a “fully continuous” model is one in which a continuum of agents changes opinions in continuous time. We note that fully continuous models in our sense were studied by Lorenz in [25]. In this paper, we derive a fully continuous version of the Hegselmann-Krause model.

We start with a time-continuous, space-discrete model. In contrast with many of the existing time-continuous models [29], we don’t interpret particles as agents, but as agent *clusters* of different sizes. Our time-continuous model has a natural space-time-continuous analogue, a differential-integral equation. The time-continuous model that we start out with can then be interpreted as a numerical method for the differential-integral equation, a particle method to be precise. We note that particle methods are a natural choice for the numerical simulation of bounded confidence models because biased assimilation tends to result in the formation of *clusters* of like-minded individuals—groups of friends confirming and equalizing each others’ opinions on Facebook or over dinner, for instance—causing accuracy issues for numerical methods based on fixed grids.

We also observe that the differential-integral equation can be translated into a system of partial differential equations without any integrals, somewhat reminiscent of the Poisson-Nernst-Planck model of electro-diffusion: The density (of individuals in opinion space, or of charged particles in physical space) moves in a velocity field that itself is determined by the density via Poisson-like differential equations. In our model, we show that an integration-by-parts argument that would lead to an *energy dissipation* inequality in physical systems leads to a *mass concentration* inequality here.

2 A Time-Continuous Model

2.1 Opinion Space and Opinion Holder Distributions

We assume that any individual’s opinions can be characterized by a single real number x . In politics, one could think of this as the “left-right axis”, with values

of x on the left side of the axis corresponding to “left” views, and values on the right side to “right” ones. This is a gross simplification that captures some aspect of the truth, since political views on different issues are correlated: If you tell us your thoughts about immigration policy and about allowing organized prayer in schools, we cannot be sure how you feel about a single-payer healthcare system, but we do have a guess.

We refer to the x -axis as “the opinion axis” or “opinion space”. A space-continuous model typically uses a time-dependent *density* of opinion holders,

$$f(x, t), \quad x \in \mathbb{R}, \quad t \geq 0. \tag{1}$$

The time t could still tick discretely in such a model, but we are primarily interested in space-time-continuous models in which t flows continuously. We always assume $f(x, t) \geq 0$, and normalize so that

$$\int_{-\infty}^{\infty} f(x, t) dx = 1 \quad \text{for all } t. \tag{2}$$

More generally and abstractly, the opinion holder distribution f could be a time-dependent Borel probability measure on \mathbb{R} ; however, the only measures without densities that we’ll talk about in this paper are weighted sums of Dirac measures.

2.2 Particle Representation of Opinion Distributions

Let $X_1, X_2, \dots, X_n \in \mathbb{R}$, and assume for now that the X_i are the only opinions represented in the electorate. If w_i is the fraction of individuals who hold opinion X_i , then the “density” of opinions altogether is the distribution

$$\sum_{i=1}^n w_i \delta(x - X_i), \tag{3}$$

where δ denotes the Dirac delta distribution. The condition that this be a probability measure becomes $\sum_{i=1}^n w_i = 1$.

Any Borel probability measure μ on the real line can be approximated arbitrarily well, in the distributional sense, by a weighted sum of delta functions in the form (3). In fact, let $m \geq 1$ be an integer, $\Delta x > 0$ a real number, and define, for all integers i with $-m + 1 \leq i \leq m - 1$,

$$w_i = \frac{\mu([(i - 1/2)\Delta x, (i + 1/2)\Delta x])}{\sum_{k=-m+1}^{m-1} \mu([(k - 1/2)\Delta x, (k + 1/2)\Delta x])}. \tag{4}$$

Then

$$\sum_{i=-m+1}^{m-1} w_i \delta(x - i\Delta x), \tag{5}$$

converges weakly to μ if $m \rightarrow \infty$ and $\Delta x \rightarrow 0$ in such a way that $m\Delta x \rightarrow \infty$.

Any weighted sum of delta functions in the form (3) can in turn be approximated arbitrarily well by a smooth density. For instance, the smooth probability density

$$\sum_{i=1}^n w_i \frac{e^{-(x-X_i)^2/(2\sigma^2)}}{\sqrt{2\pi\sigma^2}} \tag{6}$$

converges to (3), in the distributional sense, as $\sigma \rightarrow 0$.

2.3 The Dynamics of Conformist Opinion Holders

We assume that the opinion holders in the i -th cluster, that is, opinion holders with opinion X_i , consider a weighted average of opinions (including their own), in the form

$$\frac{\sum_{j=1}^n \eta(|X_i - X_j|)w_j X_j}{\sum_{j=1}^n \eta(|X_i - X_j|)w_j}$$

where

$$\eta : [0, \infty) \rightarrow [0, 1]$$

is a decreasing function with $\lim_{z \rightarrow \infty} \eta(z) = 0$, called the *interaction function*. The further X_i is removed from X_j , the less will the j -th cluster affect the opinion of the i -th cluster. We then assume that X_i drifts towards the weighted average:

$$\frac{dX_i}{dt} = \alpha \left(\frac{\sum_{j=1}^n \eta(|X_i - X_j|)w_j X_j}{\sum_{j=1}^n \eta(|X_i - X_j|)w_j} - X_i \right) \tag{7}$$

where $\alpha > 0$ is a parameter determining how eager the opinion holders are to fall in line with those who already hold opinions similar to theirs. We will take $\alpha = 1$. This is just a matter of choosing time units. Using this, and simplifying a bit, (7) becomes

$$\frac{dX_i}{dt} = \sum_{j=1}^n \frac{\eta(|X_i - X_j|)w_j}{\sum_{\ell=1}^n \eta(|X_i - X_\ell|)w_\ell} (X_j - X_i). \tag{8}$$

The model is most closely analogous to that of Hegselmann and Krause if the interaction function η is taken to be the indicator function of an interval $[0, \epsilon]$ with $\epsilon > 0$. However, we use

$$\eta(z) = e^{-z/\nu}$$

where $\nu > 0$ is a parameter determining how broad-minded the opinion holders are. Larger ν means greater broad-mindedness.

Setting $w_j = 1$ in (8), our equation simplifies to [3, Eq. (6)]. A time-discrete version of the model of [3] is also given in [12, eqs. (3) and (4)]. A time-discrete model including weights can be found for instance in [7, Eq. (2)]. If η is taken to be an indicator function, our model becomes a time-continuous version of that in [7].

2.4 Examples

We first assume that the initial opinion distribution has the density

$$f_0(x) = \frac{1}{2} \left(\frac{e^{-2(x+1)^2}}{\sqrt{\pi/2}} + \frac{e^{-2(x-1)^2}}{\sqrt{\pi/2}} \right).$$

The graph of this function is shown in Fig. 1.

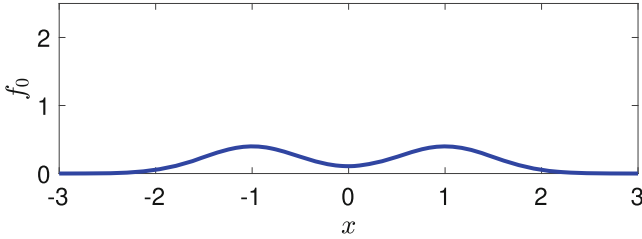


Fig. 1. An opinion distribution with two distinct “camps”, a “left” one and a “right” one.

We approximate this distribution by a weighted sum of 400 Dirac delta functions, as described by equations (4) and (5) with $m = 200$ and $\Delta x = 3/m$. We compute the time evolution as described by Eq. (8), using $\eta(z) = e^{-2z}$, using the midpoint method with $\Delta t = 0.04$. At each time $t > 0$, this results in a weighted sum of Dirac delta functions approximating the opinion distribution. We approximate this sum by a smooth probability density as defined in (6) with $\sigma = 0.1$. The upper panel of Fig. 2 shows the resulting densities at times $t = 0$ (blue), 5 (black), and 10 (red). The lower panel shows the same time evolution as a surface plot. The two initial clusters tighten, but they also move towards each other, and eventually they merge into one cluster at the center.

For the initial opinion distribution

$$f_0(x) = \frac{1}{3} \left(\frac{e^{-5(x+1)^2}}{\sqrt{\pi/5}} + \frac{e^{-5x^2}}{\sqrt{\pi/5}} + \frac{e^{-5(x-1)^2}}{\sqrt{\pi/5}} \right)$$

we obtain the time evolution shown in Fig. 3. A feature of some interest is that the three clusters start out with equal amplitude, but by time 10, the middle cluster has a lower amplitude than the outlying ones. A closer inspection of the computed density shows that this effect is mostly attributable to less tightening in the central cluster, not to migration of individuals out of the central cluster: The percentage of individuals between $x = -0.5$ and $x = 0.5$ is nearly exactly the same at time 10 as at time 0. At approximately time 30, the three clusters merge into one. This calculation was carried out with a bit less resolution: $m = 100$ (so 200 Dirac delta functions), again $\Delta x = 3/m$, and $\Delta t = 0.1$, and $\sigma = 0.1$.

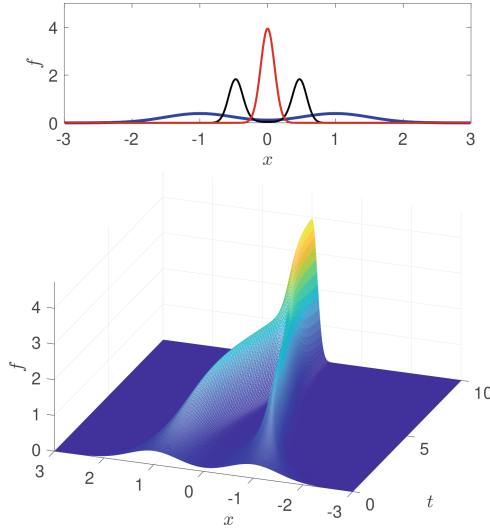


Fig. 2. If the initial opinion distribution is that shown in Fig. 1, the initial clusters tighten at first, but also move towards one another, and eventually merge. The upper panel shows f at times 0 (blue), 5 (black), and 10 (red). The lower panel shows the time evolution as a 3D surface plot.

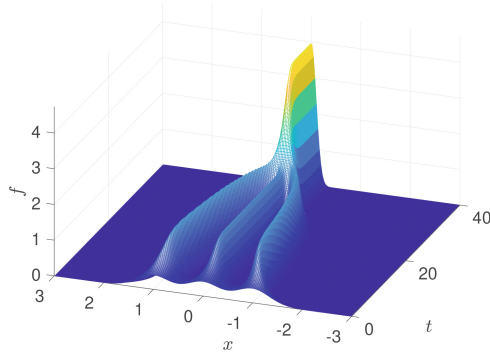


Fig. 3. Tightening and eventual collapse of three initial clusters.

2.5 Concentration of the Opinion Holder Density

We saw in our numerical results that multiple clusters in our model always appear to be transient, eventually collapsing into a single cluster. It is easy to prove that this must always happen [9, Proposition 1].

Proposition 1. *Assume that the X_i , $1 \leq i \leq n$, obey Eq. (8). Then $\min_{1 \leq i \leq n} X_i$ is increasing, $\max_{1 \leq i \leq n} X_i$ is decreasing, and*

$$\lim_{t \rightarrow \infty} \left(\max_{1 \leq i \leq n} X_i - \min_{1 \leq i \leq n} X_i \right) = 0.$$

For much less straightforward results about convergence to consensus in *higher-dimensional* spaces, see [27, Section 2].

3 A Space-Time-Continuous Model

3.1 Differential-Integral Formulation

Let now $f = f(x, t)$ be a continuous opinion holder density. The analogues of

$$\sum_{j=1}^n \eta(|X_i - X_j|) w_j X_j \quad \text{and} \quad \sum_{j=1}^n \eta(|X_i - X_j|) w_j$$

are

$$\int_{-\infty}^{\infty} \eta(|z|) f(x - z, t) (x - z) dz \quad \text{and} \quad \int_{-\infty}^{\infty} \eta(|z|) f(x - z, t) dz.$$

We will derive a continuous evolution equation using arguments similar to those often used to derive conservation equations such as convection or diffusion equations. Consider an interval $[a, b]$. At a , opinion holders are moving right with velocity

$$\frac{\int_{-\infty}^{\infty} \eta(|z|) f(a - z, t) (a - z) dz}{\int_{-\infty}^{\infty} \eta(|z|) f(a - z, t) dz} - a.$$

At b , they are moving right with velocity

$$\frac{\int_{-\infty}^{\infty} \eta(|z|) f(b - z, t) (b - z) dz}{\int_{-\infty}^{\infty} \eta(|z|) f(b - z, t) dz} - b$$

Now think about a short time interval of duration Δt . The fraction of opinion holders entering $[a, b]$ through a in the time interval $[t, t + \Delta t]$ is about

$$f(a, t) \left(\frac{\int_{-\infty}^{\infty} \eta(|z|) f(a - z, t) (a - z) dz}{\int_{-\infty}^{\infty} \eta(|z|) f(a - z, t) dz} - a \right) \Delta t.$$

The fraction exiting through b is similarly

$$f(b, t) \left(\frac{\int_{-\infty}^{\infty} \eta(|z|) f(b - z, t) (b - z) dz}{\int_{-\infty}^{\infty} \eta(|z|) f(b - z, t) dz} - b \right) \Delta t.$$

It follows that

$$\begin{aligned} \frac{d}{dt} \int_a^b f(x, t) dx &= \int_a^b f_t(x, t) dx = -f(b, t) \left(\frac{\int_{-\infty}^{\infty} \eta(|z|) f(b - z, t) (b - z) dz}{\int_{-\infty}^{\infty} \eta(|z|) f(b - z, t) dz} - b \right) + \\ &\quad f(a, t) \left(\frac{\int_{-\infty}^{\infty} \eta(|z|) f(a - z, t) (a - z) dz}{\int_{-\infty}^{\infty} \eta(|z|) f(a - z, t) dz} - a \right). \end{aligned}$$

Since this holds for any choice of $[a, b]$, we conclude:

$$f_t(x, t) = \left(\frac{\int_{-\infty}^{\infty} \eta(|z|)z f(x - z, t) dz}{\int_{-\infty}^{\infty} \eta(|z|)f(x - z, t) dz} f(x, t) \right)_x \tag{9}$$

The particle model presented in Sect. 2 can be viewed as a discretization of Eq. (9), with initial condition $f(x, 0) = f_0(x)$ and zero boundary conditions at $\pm\infty$.

Equation (9) is closely related to others that have appeared in the literature, for instance Eq. [12, eqs. (7) and (8)]. According to our convection equation (9), the velocity at opinion space location x at time t equals

$$-\frac{\int_{-\infty}^{\infty} \eta(|z|)z f(x - z, t) dz}{\int_{-\infty}^{\infty} \eta(|z|)f(x - z, t) dz} = \frac{\int_{-\infty}^{\infty} \eta(|y - x|)(y - x)f(y, t) dy}{\int_{-\infty}^{\infty} \eta(|y - x|)f(y, t) dy} \tag{10}$$

while the velocity in [12, Eq. (8)], in the same notation, is

$$\int_{-\infty}^{\infty} \eta(|y - x|)(y - x)f(y, t) dy. \tag{11}$$

In (10), a weighted average of the differences $y - x$ is taken, while (11) will be larger if x is surrounded by many nearby agents, smaller if it isn't. Which is more accurate depends on how one believes opinion dynamics to work—we assume that all agents are equally eager to conform, even those surrounded by only few other agents, whereas [12, Eq. (8)] implicitly assumes that those surrounded by many agents are more eager to conform than those surrounded by few agents.

3.2 Differential Formulation

In our numerical experiments, we always use $\eta(z) = e^{-z/\nu}$ with $\nu > 0$. This assumption was not crucial until now, but will be here; we could not do the following computation with a general η . There are two integrals in Eq. (9),

$$g(x, t) = \int_{-\infty}^{\infty} e^{-|z|/\nu} f(x - z, t) dz, \quad h(x, t) = \int_{-\infty}^{\infty} z e^{-|z|/\nu} f(x - z, t) dz. \tag{12}$$

The strategy is to write Eq. (9) as

$$f_t(x, t) = \left(\frac{h(x, t)}{g(x, t)} f(x, t) \right)_x$$

and then add supplementary differential equations for g and h . We note for later reference that

$$\int_{-\infty}^{\infty} g(x, t) dx = 2\nu \quad \text{for all } t. \tag{13}$$

In fact, $g(\cdot, t) = \eta * f(\cdot, t)$ and $\frac{\eta}{2\nu}$ is a probability density. We refer to g as the *locally averaged opinion holder density*. For later reference, we also note that

$$\int_{-\infty}^{\infty} h(x, t) dx = 0 \quad \text{for all } t. \tag{14}$$

It is not difficult to derive the following differential equations now [9]:

$$f_t = \left(\frac{h}{g} f \right)_x, \tag{15}$$

$$-g_{xx} + \frac{g}{\nu^2} = \frac{2}{\nu} f, \tag{16}$$

$$-h_{xx} + \frac{h}{\nu^2} = -2g_x. \tag{17}$$

3.3 Concentration of the Locally Averaged Opinion Holder Density

We define

$$H(x, t) = \int_{-\infty}^x h(s, t) ds.$$

Note $H_x = h$ and $H(-\infty, t) = 0$. Integrating Eq. (17), we obtain

$$-H_{xx} + \frac{H}{\nu^2} = -2g + C$$

with C independent of x . Assuming that $H_{xx} = h_x$ and g vanish at $x = -\infty$, and using Eq. (14), we conclude $C = 0$, so

$$-H_{xx} + \frac{H}{\nu^2} = -2g. \tag{18}$$

Multiply both sides of (15) by H , integrate with respect to x , and then integrate by parts on the right-hand side:

$$\int_{-\infty}^{\infty} H(x, t) f_t(x, t) dx = - \int_{-\infty}^{\infty} \frac{h^2(x, t)}{g(x, t)} f(x, t) dx. \tag{19}$$

On the left-hand side of this equation, now use Eq. (16):

$$\int_{-\infty}^{\infty} H(x, t) f_t(x, t) dx = \frac{\nu}{2} \int_{-\infty}^{\infty} H(x, t) \left(-g_{xx} + \frac{g}{\nu^2} \right)_t dx$$

Integrate by parts twice:

$$\frac{\nu}{2} \int_{-\infty}^{\infty} H(x, t) \left(-g_{xx} + \frac{g}{\nu^2} \right)_t dx = \frac{\nu}{2} \int_{-\infty}^{\infty} \left(-H_{xx} + \frac{1}{\nu^2} H \right) g_t dx.$$

Now use Eq. (18):

$$\frac{\nu}{2} \int_{-\infty}^{\infty} \left(-H_{xx} + \frac{1}{\nu^2} H \right) g_t dx = -\frac{\nu}{2} \int_{-\infty}^{\infty} 2gg_t dx = -\frac{\nu}{2} \frac{d}{dt} \|g\|_{L^2}.$$

Using this in Eq. (19), we find:

$$\frac{d}{dt} \|g\|_{L^2}^2 = \frac{2}{\nu} \int_{-\infty}^{\infty} \frac{h^2(x, t)}{g(x, t)} f(x, t) dx. \tag{20}$$

Positivity of f implies positivity of g , and therefore

$$\frac{d}{dt} \|g\|_{L^2}^2 > 0. \tag{21}$$

Recall from Eq. (13) that the L^1 -norm of g is equal to 2ν for all time. The square of the L^2 -norm is a measure of concentration of g . This is reflected by the fact that if X is a random number with probability density $\frac{g}{2\nu}$, then $\|g\|_{L^2}^2 = 2\nu E(g(X))$, and therefore $E(g(X))$ rises as t increases. When X is drawn with density g , the expected value of $g(X)$ gets larger with time. This means that g becomes increasingly concentrated.

4 Numerical Convergence Tests

As an example, we test convergence for the initial condition

$$f_0(x) = \frac{1}{3} \left(\frac{e^{-5(x+1)^2}}{\sqrt{\pi/5}} + \frac{e^{-5x^2}}{\sqrt{\pi/5}} + \frac{e^{-5(x-1)^2}}{\sqrt{\pi/5}} \right).$$

We track the X_j up to time $t = 1$. We use a time step Δt and assume that $1/\Delta t$ is an integer. We initialize the X_j at

$$X_j(0) = -3 + j\Delta x, \quad j = 1, 2, \dots, \frac{6}{\Delta x} - 1,$$

assuming that $6/\Delta x$ is an integer. We compute approximations for $f(x, t)$, $x = j\Delta x$, j integer, Eq. (6), where $\sigma = 0.1$. We denote these approximations by $f_{\Delta x, \Delta t}(x, t)$, and will test whether they converge to *some* limit as Δx and Δt are simultaneously reduced. We cannot test convergence to an exact solution, since we have no analytic expression for an exact solution.

Fixing σ independently of Δx amounts to testing for a form of *weak* convergence. The computed distribution is a sum of δ -functions, but we test for convergence of the convolution with a Gaussian.

We define

$$E_{\Delta x, \Delta t} = \max \left\{ \left| f_{\frac{\Delta x}{2}, \frac{\Delta t}{2}}(x, 1) - f_{\Delta x, \Delta t}(x, 1) \right| : \frac{x}{\Delta x} \text{ integer, } -3 < x < 3 \right\}.$$

If there is second-order convergence as Δx and Δt simultaneously tend to zero, one should expect

$$\frac{E_{\Delta x, \Delta t}}{E_{\Delta x/2, \Delta t/2}} \approx 4$$

for small Δx and Δt . Table 1 confirms that this is indeed the case.

More extensive convergence tests, confirming weak second order convergence, are presented in [9].

Table 1. Numerical test confirming second order convergence of the approximation obtained by convolving the computed sum of delta functions with a Gaussian, as both Δx and Δt are refined; see text for details.

Δx	0.06	0.03	0.015
Δt	0.1	0.05	0.025
$E_{\Delta x, \Delta t} / E_{\Delta x/2, \Delta t/2}$	4.01	3.98	4.00

5 Summary and Discussion

We began with a time-continuous version of Hegselmann-Krause dynamics, very similar to equations that have been proposed in the literature previously, but with weighted particles, which we think of as representing clusters of agents, not individuals. The weights have a numerical advantage—instead of needing many agents in a part of opinion space populated by many opinion holders, we can use fewer but heavier particles.

The time-continuous model suggests a fully continuous macroscopic model, which we formulated first as a single integral-differential equation, then—for the special case of an exponential interaction function—as a system of differential equations. The time-continuous model (discretized using the midpoint method) can be viewed as a particle method for the fully continuous model.

In our numerical computations, all opinion holders eventually arrive at consensus. This could be counter-acted by adding *diffusion* (spontaneous random small changes in opinions) in the model, as some authors have proposed (see for instance [4, 16]). We have refrained from doing that here because it would raise, in our context, the question how to incorporate diffusion in the particle method. One possibility would be a method similar to Chorin’s random walk method for viscous fluid dynamics [14, 17].

In future work, we plan to use the method presented in this paper to explore the interaction of candidate dynamics with voter opinion dynamics. We have taken a first step in that direction in [8].

Another avenue for further exploration could come from considering the interaction function to be additionally dependent on agents’ properties that are unrelated to their opinion, such as connections in a social network or geographic location [10]. The former is closely related to targeted advertising research. Additionally, one could introduce dependence of the interaction function on other forms of opinion average (such as geometric average) as suggested by Hegselmann and Krause in [19].

Acknowledgments. This work was supported by a grant from the Data Intensive Studies Center at Tufts University. We thank Bruce Boghosian for helpful discussions.

References

1. Acemoglu, D., Ozdaglar, A.: Opinion dynamics and learning in social networks. *Dyn. Games Appl.* **1**, 3–49 (2011)
2. Anderson, B.D.O., Ye, M.: Recent advances in the modelling and analysis of opinion dynamics on influence networks. *Int. J. Autom. Comput.* **16**, 129–149 (2019)
3. Aydoğdu, A., et al.: Interaction network, state space, and control in social dynamics. In: Bellomo, N., Degond, P., Tadmor, E. (eds.) *Active Particles*, Volume 1. MSSET, pp. 99–140. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-49996-3_3
4. Ben-Naim, E.: Rise and fall of political parties. *Europhys. Lett.* **69**, 671–677 (2005)
5. Bisin, A., Verdier, T.: Beyond the melting pot: cultural transmission, marriage, and the evolution of ethnic and religious traits. *Q. J. Econ.* **115**, 955–988 (2000)
6. Bisin, A., Verdier, T.: The economics of cultural transmission and the dynamics of preferences. *J. Econ. Theory* **97**, 298–319 (2001)
7. Blondel, V.D., Hendrickx, J.M., Tsitsiklis, J.N.: On Krause’s multi-agent consensus model with state-dependent connectivity. *IEEE Trans. Autom. Control* **54**, 2586–2597 (2009)
8. Börgers, C., Boghosian, B., Dragovic, N., Haensch, A.: The perils of centrism. Preprint accepted by *The American Mathematical Monthly* (2023)
9. Boghosian, B., Börgers, C., Boghosian, B., Dragovic, N., Haensch, A., Kirshtein, A.: A particle method for continuous Hegselmann-Krause Opinion Dynamics. arXiv preprint [arXiv:2211.06265](https://arxiv.org/abs/2211.06265) (2022)
10. Boghosian, B., Börgers, C., Boghosian, B., Dragovic, N., Haensch, A.: A geospatial bounded confidence model including mega-influencers with an application to COVID-19 vaccine hesitancy. *JASSS* (2023). <https://www.jasss.org/26/1/8.html>
11. Boyd, R., Richerds, P.: *Culture and The Evolutionary Process*. The University of Chicago Press, Chicago (1985)
12. Canuto, C., Fagnani, F., Tilli, P.: An Eulerian approach to the analysis of Krause’s consensus models. *SIAM J. Control. Optim.* **50**, 243–265 (2012)
13. Cavalli-Sforza, L., Feldman, M.: *Cultural Transmission and Evolution: A Quantitative Approach*. Princeton University Press, Princeton (1981)
14. Chorin, A.J.: Numerical study of slightly viscous flow. *J. Fluid Mech.* **57**, 785–796 (1973)
15. Fortunato, S., Latora, V., Pluchino, A., Rapisarda, A.: Vector opinion dynamics in a bounded confidence consensus model. *Int. J. Modern Phys. C* **16**, 1535–1551 (2005)
16. Goddard, B.D., Gooding, B., Short, H., Pavliotis, G.A.: Noisy bounded confidence models for opinion dynamics: the effect of boundary conditions on phase transitions. *IMA J. Appl. Math.* **87**, 80–110 (2022)
17. Goodman, J.: Convergence of the random vortex method. In: Papanicolaou, G. (eds.) *Hydrodynamic Behavior and Interacting Particle Systems*. The IMA Volumes in Mathematics and Its Applications, vol. 9. Springer, New York (1987). https://doi.org/10.1007/978-1-4684-6347-7_7
18. Hegselmann, R., Krause, U.: Opinion dynamics and bounded confidence models, analysis, and simulation. *JASSS* **5**(3), 1–2 (2002)
19. Hegselmann, R., Krause, U.: Opinion dynamics driven by various ways of averaging. *Comput. Econ.* **25**, 381–405 (2005)
20. Krause, U.: Soziale Dynamiken mit vielen Interakteuren. Eine Problemskizze, Modellierung und Simulation von Dynamiken mit vielen interagierenden Akteuren, 3751(2), 3751–3753 (1997)

21. Krause, U.: A discrete nonlinear and non-autonomous model of consensus formation. *Commun. Differ. Equ.* **2000**, 227–236 (2000)
22. Lord, C.G., Ross, L., Lepper, M.R.: Biased assimilation and attitude polarization: the effects of prior theories on subsequently considered evidence. *J. Pers. Soc. Psychol.* **37**, 2098–2109 (1979)
23. Lorenz, J.: A stabilization theorem for dynamics of continuous opinions. *Physica A* **355**, 217–223 (2005)
24. Lorenz, J.: Consensus strikes back in the Hegselmann-Krause model of continuous opinion dynamics under bounded confidence. *JASSS* (2006). <https://www.jasss.org/9/1/8.html>
25. Lorenz, J.: Continuous opinion dynamics under bounded confidence: a survey. *Int. J. Modern Phys. C* **18**(12), 1819–1838 (2007)
26. Mirtabatabaei, A., Bullo, F.: Opinion dynamics in heterogeneous networks: convergence conjectures and theorems. *SIAM J. Control Optim.* **50**(5), 2763–2785 (2012)
27. Montsch, S., Tadmor, E.: Heterophilious dynamics enhance consensus. *SIAM Rev.* **56**, 577–621 (2014)
28. Mossel, E., Tamuz, O.: Opinion exchange dynamics. *Probab. Surv.* **14**, 155–204 (2017)
29. Piccoli, B., Rossi, F.: Generalized solutions to bounded-confidence models. *Math. Models Methods Appl. Sci.* **31**(06), 1237–1276 (2021)
30. Proskurnikov, A.V., Tempo, R.: A tutorial on modeling and analysis of dynamic social networks. Part I. *Annu. Rev. Control* **43**, 65–79 (2017)
31. Proskurnikov, A.V., Tempo, R.: A tutorial on modeling and analysis of dynamic social networks. Part II. *Annu. Rev. Control* **45**, 166–190 (2018)
32. Urbig, D.: Attitude dynamics with limited verbalisation capabilities. *J. Artif. Societies Social Simul.* **6**(1), 2 (2003)
33. Wedin, E., Hegarty, P.: The Hegselmann-Krause dynamics for the continuous-agent model and a regular opinion function do not always lead to consensus. *IEEE Trans. Autom. Control* **60**, 2416–2421 (2015)
34. Weissbuch, G., Deffuant, G., Amblard, F., Nadal, J.-P.: Meet, discuss, and segregate! *Complexity* **7**, 55–63 (2002)



Optimal Reconstruction of Graph Evolution Dynamics for Duplication-Based Models

Emre Sefer^{1(✉)} and Samuel Gilmour²

¹ Computer Science Department, Ozyegin University, Istanbul, Turkey
emre.sefer@ozyegin.edu.tr

² MIT Operations Research Center, Massachusetts, USA

Abstract. The research on biological network evolution and graph growth models, such as the Duplication-Mutation with Random Mutation (DMR) model, enable us to characterize the protein interaction network's evolutionary dynamics founded on duplication and divergence via mutation in a principled way. The existing approaches to reconstruct historical ancestral graphs for DMR model mainly focus on greedy approaches and results in suboptimal solutions. In this study, we come up with ILP-DMR, a novel Integer Linear Programming (ILP)-based formulation, to reconstruct historical PPI graphs by likelihood maximization over DMR model. We assess the effectiveness of our approach in reconstructing the history of synthetic as well as optimal history of the proteins from the families of bZIP transcription factors. In comparison to the existing techniques, solutions returned by our ILP-DMR have a higher likelihood and are more robust to model mismatch and noise in the data. Solutions extracted by ILP-DMR have a higher likelihood than the existing methods, and our solutions better agree with the biological findings of different studies. Our datasets and code are available at <https://github.com/seferlab/dmrhistory>.

Keywords: DMR · Duplication-Mutation with Random Mutation · Ancestral History Reconstruction · PPI

1 Introduction

Graph growth models, a type of generative models, express and model the graph evolution over time, and they have been utilized to explain graph attributes in distinct domains. Some examples are Forest Fire Model [10] for social graphs and Preferential Attachment Model [2] mainly to model WWW. Those growth models model the evolutionary dynamics assumptions by graph operations. Duplication and mutation is the key evolutionary procedure governing the growth of biological networks [22]. In this case, a graph vertex is duplicated involving its incident edges which is followed by the deletion of a number of the incident edges, and these two operations as a whole model a single evolutionary time step.

A number of biological studies have validated these models [20]. In this study, we focus on Duplication-Mutation with Random Mutation (DMR) model, that outperforms the remaining models in fitting PPI graphs in terms of various aspects, such as the -hop reachability (i.e., the number of distinct nodes that can be reached from a given node via a path of edges), the graphlet distribution, as well as the betweenness, closeness, and degree distributions [3, 12, 14].

One can reconstruct the historical networks in a generative model-based principled way by using a graph growth model, similar to inferring the evolutionary history of sequences via reconstruction algorithms. Under the assumption of a generative model, history graph reconstruction focuses on finding the graph sequence with the highest likelihood yielding the present network. Such ancestral reconstruction requires the vertex order inference in which order vertices in the graph duplicate and the graph lose interactions between vertices at every individual step throughout evolution. A number of methods have been developed for historical graph reconstruction, especially for DMC model instead of DMR model. For instance, [14] has come up with a greedy ReverseDMC algorithm that focuses on reconstructing the network history with the highest likelihood over DMC model. This ReverseDMC method greedily maximizes the likelihood of every individual step of evolution and selects an anchor vertex which is duplicated at every step. Such ReverseDMC algorithm focuses solely on the current graph's topology to reconstruct ancestral historical graphs. A number of its variants have been introduced, which integrate extra biological knowledge about the proteins into the inference. For example, [7] integrates vertex contents over ancestral graphs as the additional knowledge into ancestral inference. Similarly, such extra knowledge could be duplication history of proteins [6, 11]. There are additional ancestral graph inference algorithms that are based on minimum-parsimony principle [15]. A number of other techniques infer histories over Preferential Attachment Model [19, 23]. Recently, [17] has proposed an integer linear programming-based solution to optimally infer ancestral PPI network histories using DMC model. Almost all of these approaches extract solely a single evolutionary history via optimization criteria such as likelihood. Almost none of them except [17] can infer optimal evolutionary histories due to their formulations.

In this study, we come up with ILP-DMR, an Integer Linear Programming (ILP) solution, to optimally reconstruct historical PPI networks for DMR model by maximizing the likelihood, and by taking into account only the existing graph's knowledge. We define indicator variables to model duplicated and anchor vertices at every evolutionary step. We integrate DMR model's conditions as linear constraints for every successive pair of graphs as part of evolution to the optimization. We evaluate our proposed ILP-based method in synthetic and real datasets. In synthetic data, our method outperforms greedy ReverseDMR, showing the non-optimality of the greedy heuristic. Moreover, we compare ILP-DMR with ReverseDMR via different metrics such as similarity of correct graph with the inferred one (as evaluated by similarity via graph kernel), vertex arrival order (evaluated by Kendall's Tau metric). In terms of all those metrics, ILP-based reconstructions from ILP-DMR are better than the ones from ReverseDMR.

We have also assessed the performance over PPI of bZIP transcription factors families by using the same evaluation metrics. The solution obtained via ILP-DMR has a higher likelihood than the greedy method. In terms of biological evaluations, reconstructions by ILP-DMR agree more remarkably with the biological knowledge from protein sequence similarity and gene orthologs.

2 Problem Formulation

Let G_t be a graph at time t and I be an evolutionary model which defines a list of operations generating G_{g+1} from G_g . Then, we are interested in finding the most plausible evolutionary graph sequence $G_S = G_1, \dots, G_{t-1}$:

$$G_S^* = \underset{G_S}{\operatorname{argmax}}(P(G_t|G_{t-1}) \dots P(G_2|G_1) P(G_1)) \tag{1}$$

where $P(G_t|G_{t-1})$ probabilities are dependent upon the present-day graph G_t and the model l . Next, we discuss the way of likelihood computation for DMR.

According to DMR model, G_2 is a graph with 2 vertices that has an edge between them. q_{del} and q_{mod} are its parameters, and graph evolution dynamics at any step from G_g to G_{g+1} is as follows:

1. DMR randomly chooses an anchor vertex u in G_g and duplicates it to generate new vertex v . Duplicated v is exactly a copy of u : It has an edge with all u 's neighbours, and does not have an edge to the remaining vertices.
2. DMR randomly removes the edges between u and v with probability q_{del} .
3. For each neighbour x 's edges, DMR randomly modifies the either connecting edges (u, x) or (v, x) with q_{mod} probability; If the edge is selected for modification, DMR eliminates either (u, x) or (v, x) with identical probability.

G_g represents a graph having g vertices since DMR adds a single vertex at every evolutionary step. Let $e_{uv} \in \{0, 1\}$ define the existence of edge between anchor vertex u and duplicated vertex v . According to DMR model's second step above, $(1 - q_{\text{mod}})$ is the probability of u and v having a common neighbour, $q_{\text{mod}}/2$ is the probability that a vertex x is a neighbour of u and not of v , or it is a neighbour of v and not of u . Let $N(u)$ be the list of u 's neighbours, $N(u) \cap N(v)$ be the list of shared neighbours of u and v , and $N(u) \Delta N(v)$ be the list of vertices which are neighbours of either u or v but not of both. Subsequently, provided with anchor u and duplicated v , log-likelihood $\log P(G_g|G_{g-1}, l)$ is defined as:

$$\log P(G_g|G_{g-1}, l) = e_{uv} \log(1 - q_{\text{del}}) + (1 - e_{uv}) \log(q_{\text{del}}) + \sum_{N(u) \cap N(v)} \log(1 - q_{\text{mod}}) + \sum_{N(u) \Delta N(v)} \log(q_{\text{mod}}) \tag{2}$$

where constant terms are ignored. Immediately after u and v are determined, one can reconstruct G_{g-1} via eliminating duplicated vertex v and attaching back the edges between anchor vertex u and each vertex in $N(u) - (N(u) \cap N(v))$.

In this case, these edges are attached back since they already existed before DMR’s second step has been applied. One should keep in mind that u and v are not distinguishable in G_g : While reconstructing back G_{g-1} , either u or v can be eliminated and corresponding edge additions follow mutatis mutandis. From now on, the vertex pair u, v in G_g will be referred to as duplicated vertices and u in G_{g-1} as the anchor vertex.

3 ILP-DMR: Integer Linear Programming-Based Solution

3.1 Our ILP

Given the present-day network G_t , we should infer the succeeding knowledge to reconstruct the whole series of graphs G_S :

1. Anchor vertices in all previous networks G_2, \dots, G_{t-1}
2. Duplicated vertices in all networks G_3, \dots, G_t
3. Edges of all networks G_3, \dots, G_{t-1}

We reconstruct the evolutionary history of a graph by formulating the solution as an ILP. We define i_g as i ’th node in graph g and will eliminate the subscript in variable namings once the corresponding graph is apparent, to prevent confusion. For every graph G_2, \dots, G_t , we will model the existence of an edge by binary edge indicators e_{ijg} , and binary vertex indicators $x_{ig}, y_{ig}, z_{ig}, a_{ig}$. Subscripts i and j denote the vertices and g denote the graph G_g having vertices $1, \dots, g$. When i th vertex of G_g is a duplicated/anchor vertex, x_{ig}/a_{ig} will be assigned 1 respectively. Indicator variable y_{ig} will be used to recognize duplicated vertices shared neighbour. Similarly, indicator variable z_{ig} will be used to recognize a neighbour of either of the duplicated vertices but not of both. Among edge indicators, $e_{ijg}, \forall i, j$ are predetermined for graphs G_2, G_t and are not known in the remaining graphs. However, none of the vertex indicators are known for graphs. In our case, we can express the log-likelihood in Eq. 1 as:

$$\begin{aligned}
 ILP = \sum_{g=1}^t & \left(\sum_{i=1}^g \sum_{j=1}^g e_{ijg} x_{ig} x_{jg} \log(1 - q_{\text{del}}) + (1 - e_{ijg}) x_{ig} x_{jg} \log(q_{\text{del}}) \right) \\
 & + \sum_{k=1}^g y_{kg} \log(1 - q_{\text{mod}}) + \sum_{k=1}^g z_{kg} \log q_{\text{mod}} \tag{3}
 \end{aligned}$$

Therefore, maximizing ILP term subject to all constraints ((4)–(25)) defined below is our goal where constraints are defined over the present-day graph and DMR model.

3.2 Anchors, Duplicated Nodes and Neighbours

Every graph in the problem, apart from the initial one G_2 , owns 2 duplicated vertices:

$$\sum_{i=1}^g x_{ig} = 2, \quad \forall g \in 3, \dots, t \tag{4}$$

Every graph in the problem, apart from the last one G_t , owns a single anchor vertex:

$$\sum_{i=1}^g a_{ig} = 1, \quad \forall g \in 2, \dots, t - 1 \tag{5}$$

The term $e_{ijg}x_{ig}$ becomes 1 only when i is a duplicated vertex and an edge exists from vertex j to i .

If vertex k is a shared neighbour, absolutely 2 edges must exist to the duplicated vertices in the graph. As 2 duplicated vertices exist for each graph, the summation term $\sum_{i=1}^g e_{ikg}x_{ig}$ may only get three values $\{0, 1, 2\}$ for k th vertex. When this expression is either 0 or 1, $y_{kg} = 0$ is set by constraint (6). When it is 2, $y_{kg} = 1$ is set by constraints (6) and (7).

$$2y_{kg} \leq \sum_{i=1}^g e_{ikg}x_{ig}, \quad \forall k, \forall g \in 3, \dots, t \tag{6}$$

$$y_{kg} \geq \sum_{i=1}^g e_{ikg}x_{ig} - 1, \quad \forall k, \forall g \in 3, \dots, t \tag{7}$$

Only a single edge towards the duplicated vertices shall exist in the graph, to spot a neighbour of only one of the duplicated vertices which corresponds to setting $z_{kg} = 1$. In this case, one of the duplicated vertices, which can also satisfy this criterion when an edge exists between the duplicated vertices, must not be chosen. Those constraints could be posed by defining an extra binary vertex variable w_{kg} :

$$w_{kg} + 2y_{kg} = \sum_{i=1}^g e_{ikg}x_{ig}, \quad \forall k, \forall g \in \{3, \dots, t\} \tag{8}$$

$$z_{kg} \geq w_{kg} - x_{kg}, \quad \forall k, \forall g \in \{3, \dots, t\} \tag{9}$$

$$z_{kg} \leq w_{kg}, \quad \forall k, \forall g \in \{3, \dots, t\} \tag{10}$$

$$z_{kg} \leq 1 - x_{kg}, \quad \forall k, \forall g \in \{3, \dots, t\} \tag{11}$$

As mentioned above, the term $\sum_{i=1}^g e_{ikg}x_{ig}$ can only get three values $\{0, 1, 2\}$ for k th vertex:

- When $\sum_{i=1}^g e_{ikg}x_{ig} = 2$, $y_{kg} = 1$ is ensured by constraints (6) and (7), and $w_{kg} = 0$ is set by constraint (8) resulting in $z_{kg} = 0$ via constraint (10).

- When $\sum_{i=1}^g e_{ikg}x_{ig} = 1$, constraints (6) and (7) guarantee $y_{kg} = 0$ so $w_{kg} = 1$. In such setting, constraint (11) guarantees $z_{kg} = 0$ when $x_{kg} = 1$. Similarly, constraint (9) guarantees $z_{kg} = 1$ when $x_{kg} = 0$.
- When $\sum_{i=1}^g e_{ikg}x_{ig} = 0$, constraints (6)–(8) ensure $w_{kg} = 0$ and constraint (10) ensures $z_{kg} = 0$.

We also define a different binary vertex variable n_{kg} to stress a duplicated vertex’s neighbour, where it can be a shared neighbour as well as neighbour of only one of the duplicated vertices:

$$n_{kg} = y_{kg} + z_{kg}, \quad \forall k, \forall g \in \{3, \dots, t\} \tag{12}$$

3.3 Phantom Edges

As part of reconstructing history, we should determine how G_g ’s vertices correspond to G_{g-1} ’s vertices at an earlier graph for setting the values of the unknown edges. We are especially interested in associating G_g ’s duplicated vertices with G_{g-1} ’s anchor vertex. We define indicators $P_{j_g}^{i_{g-1}}$ for pairs of vertices (i_{g-1}, j_g) to properly associate duplicated vertices with anchors, where the subscripts note the graphs to which the vertices belong to. We name the edges corresponding to these indicators as *phantom edges*, as such edges do not originally exist in the graph but instead are artificially constructed for our history reconstruction purposes. They can be viewed as directed edges from a graph at a previous time step to a current graph.

For each vertex j_g over a graph, only a single incoming phantom edge exists from any vertices in the previous network (i_{g-1}) to j_g except over G_2 :

$$\sum_{i_{g-1}=1}^{g-1} P_{j_g}^{i_{g-1}} = 1, \quad \forall j_g, \forall g \in \{3, \dots, t\} \tag{13}$$

The number of outgoing phantom edges for every vertex i_{g-1} in the previous graph is between 1 and 2, except the last graph G_t . In this case, 2 phantom edges exist for anchor vertices and only a single vertex exists for the remaining vertices:

$$\sum_{i_g=1}^g P_{j_g}^{i_{g-1}} \geq 1, \quad \forall i_{g-1}, \forall g \in \{3, \dots, t\} \tag{14}$$

$$\sum_{i_g=1}^g P_{j_g}^{i_{g-1}} \leq 2, \quad \forall i_{g-1}, \forall g \in \{3, \dots, t\} \tag{15}$$

3.4 Edge Reconstruction

We formulate the final constraints for all ancestral graph edges defined by DMR model and edges of the present-day graph. We map the edges from G_g to G_{g-1} by using phantom edges. In this case, we map all edges in the present-day graph backwards until the initial graph G_2 by ensuring the subsequent conditions:

1. We will not map an edge between duplicated vertices to any edge in previous step's graph as duplicated vertices are derived from just one anchor node.
2. We will map an edge (x_g, n_g) between a duplicated vertex x_g and its neighbour n_g in graph G_g to an edge (a_{g-1}, n_{g-1}) between the anchor a_{g-1} and its neighbour n_{g-1} in G_{g-1} .
3. We will map any remaining edge back to a unique edge in previous step graph, so no remaining edges in previous step's graph should exist.

We define three sets of variables to define constraints from these conditions. Firstly, we define a binary indicator S_{ijg}^1 for vertices i_g and j_g in G_g as follows:

$$S_{ijg}^1 = \sum_{k=1}^g a_{k(g-1)} P_{i_g}^{k_{g-1}} P_{j_g}^{k_{g-1}}$$

$S_{ijg}^1 = 1$ only when 2 phantom edges exist from an anchor vertex k_{g-1} in G_{g-1} to the vertices i_g and j_g in G_g . For every edge (i, j) , every term in the summation is multiplication of $a_{k(g-1)}$, $P_{i_g}^{k_{g-1}}$, and $P_{j_g}^{k_{g-1}}$. Such term's value becomes 1 only when the components $a_{k(g-1)} = P_{i_g}^{k_{g-1}} = P_{j_g}^{k_{g-1}} = 1$ which maps vertices i, j to the anchor vertex in the previous step graph.

S_{ijg}^{2a} is the second binary indicator over 2 vertices i_g and j_g in G_g , which is defined as:

$$S_{ijg}^{2a} = \sum_{l,k=1}^{g-1} a_{l(g-1)} (1 - a_{k(g-1)}) P_{i_g}^{k_{g-1}} P_{j_g}^{l_{g-1}} e_{lk(g-1)}$$

$S_{ijg}^{2a} = 1$ only when 2 phantom edges connect an anchor vertex $a_{l(g-1)}$ and its neighbour $(1 - a_{k(g-1)})$ respectively to i_g and j_g in G_g , and an edge $e_{lk(g-1)}$ exists in G_{g-1} . Similar to S_{ijg}^{2a} , we propose a different binary indicator S_{ijg}^{2b} for 2 nodes i_g and j_g in G_g . In this case, $S_{ijg}^{2b} = 1$ only when 2 phantom edges connect an anchor vertex $a_{l(g-1)}$ and its neighbour $(1 - a_{k(g-1)})$ respectively to j_g and i_g in G_g .

$$S_{ijg}^{2b} = \sum_{l,k=1}^{g-1} a_{l(g-1)} (1 - a_{k(g-1)}) P_{i_g}^{k_{g-1}} P_{j_g}^{l_{g-1}} e_{lk(g-1)}$$

We map from vertices i and j to an anchor vertex and its neighbour over the previous step graph by using each summation term in S_{ijg}^{2a} and S_{ijg}^{2b} .

Lastly, binary indicator T_{ijg} for two vertices i_g and j_g in G_g is defined as:

$$T_{ijg} = \sum_{l,k=1}^{g-1} P_{i_g}^{l_{g-1}} P_{j_g}^{k_{g-1}} e_{lk(g-1)}$$

$T_{ijg} = 1$ only when 2 phantom edges exist from (any) vertices k_{g-1} and l_{g-1} in G_{g-1} to i_g and j_g in G_g respectively, and an edge $e_{lk(g-1)}$ also exists. Each summation term is a multiplication of phantom vertices incoming at i and j in

G_g , and the edge $e_{lk(g-1)}$ in previous step's graph G_{g-1} . Each term maps from edge $(i, j) \in G_g$ to edge $(l, k) \in G_{g-1}$, when it is 1.

We define the conditions for constraints for each vertex pair (i_g, j_g) in graph G_g , which depend on vertex indicators for duplicated vertices (x_{i_g}) and neighbour vertices (n_{i_g}):

- If $x_{i_g}x_{j_g} = 1$ meaning that both i_g and j_g are duplicated vertices, $S_{i_jg}^1 = 1$ to enforce duplicated vertices to be connected to an anchor vertex in the previous step's graph. The remaining indicators $S_{i_jg}^{2a} = S_{i_jg}^{2b} = T_{i_jg} = 0$ to enforce that none of the edges in G_{g-1} can be mapped to an edge, if any, between vertices i_g and j_g .
- If $x_{i_g}n_{i_g} = 1$ or $x_{j_g}n_{i_g} = 1$ meaning that one of the vertices i_g and j_g is a duplicated vertex and the second one is a neighbour, $S_{i_jg}^1 = 0$ so the anchor vertex in the previous step's graph cannot be connected to that vertex pair via any phantom edges. Next assignments are $S_{i_jg}^{2b} = x_{j_g}n_{i_g}$ and $S_{i_jg}^{2a} = x_{i_g}n_{j_g}$ which enforces anchor vertex and its neighbour in the previous step's graph to be connected to nodes i_g and j_g via phantom edges. If j_g is a neighbour to the other duplicated vertex and not to i_g , an edge may not exist between i_g and j_g . That constraint is set as required, as both duplicated vertices map to the anchor. $T_{i_jg} = 1$ enforces a just single edge existence between (l_{g-1}, k_{g-1}) and $T_{i_jg} \geq e_{i_jg}$ as an edge between i_g and j_g might or might not exist. There is a mutual exclusion between this and previous cases as we never assign both $n_{i_g} = 1$ and $x_{i_g} = 1$ for the same vertex.
- If $x_{i_g}x_{j_g} = x_{i_g}n_{i_g} = x_{j_g}n_{i_g} = 0$, then $S_{i_jg}^1 = S_{i_jg}^{2b} = S_{i_jg}^{2a} = 0$ as an edge between i_g and j_g should not map to any edge connecting to an anchor in the previous step's graph. Additionally, $T_{i_jg} = e_{i_jg}$ to enforce the existence of a single edge (l_{g-1}, k_{g-1}) if $e_{i_jg} = 1$. On the other hand, if $e_{i_jg} = 0$, none of the edges in the previous graph can be mapped to (i_g, j_g) .

The following constraints incorporate the three conditions discussed above:

$$S_{i_jg}^1 = x_{i_g}x_{j_g}, \quad \forall i_g, \forall j_g, \forall g \in \{3, \dots, t\} \quad (16)$$

$$S_{i_jg}^{2a} = x_{i_g}n_{j_g}, \quad \forall i_g, \forall j_g, \forall g \in \{3, \dots, t\} \quad (17)$$

$$S_{i_jg}^{2b} = x_{j_g}n_{i_g}, \quad \forall i_g, \forall j_g, \forall g \in \{3, \dots, t\} \quad (18)$$

Additional binary variable $P_{i_jg} = 0$ if $S_{i_jg}^{2a} = 0$ and $S_{i_jg}^{2b} = 0$. Otherwise, $P_{i_jg} = 1$. Additionally, $Q_{i_jg} = x_{i_g}x_{j_g}$:

$$P_{i_jg} \geq S_{i_jg}^{2a}, \quad \forall i_g, \forall j_g, \forall g \in \{3, \dots, t\} \quad (19)$$

$$P_{i_jg} \geq S_{i_jg}^{2b}, \quad \forall i_g, \forall j_g, \forall g \in \{3, \dots, t\} \quad (20)$$

$$P_{i_jg} \geq S_{i_jg}^{2a} + S_{i_jg}^{2b}, \quad \forall i_g, \forall j_g, \forall g \in \{3, \dots, t\} \quad (21)$$

We set T_{i_jg} and e_{i_jg} by using P_{i_jg} and Q_{i_jg} :

$$T_{i_jg} \geq P_{i_jg}, \quad \forall i_g, \forall j_g, \forall g \in \{3, \dots, t\} \quad (22)$$

$$T_{ijg} \leq 1 + P_{ijg} - Q_{ijg}, \quad \forall i_g, \forall j_g, \forall g \in \{3, \dots, t\} \quad (23)$$

$$e_{ijg}(1 - Q_{ijg}) \leq T_{ijg}(1 - Q_{ijg}), \quad \forall i_g, \forall j_g, \forall g \in \{3, \dots, t\} \quad (24)$$

$$e_{ijg}(1 - P_{ijg}) \leq T_{ijg}(1 - P_{ijg}), \quad \forall i_g, \forall j_g, \forall g \in \{3, \dots, t\} \quad (25)$$

- If $Q_{ijg} = x_{ig}x_{jg} = 1$, constraints (23)–(24) enforce $T_{ijg} = P_{ijg} = 0$ since $S_{ijg}^{2a} = 0$ and $S_{ijg}^{2b} = 0$. If $P_{ijg} = 0$ and $Q_{ijg} = 1$, constraint (24) becomes void and constraint (25) enforces $e_{ijg} \geq T_{ijg}$.
- If $Q_{ijg} = x_{ig}x_{jg} = 0$ and $P_{ijg} = 1$, then either $S_{ijg}^{2a} = 1$ or $S_{ijg}^{2b} = 1$ which is only possible when $x_{ig}n_{ig} = 1$ or $x_{jg}n_{jg} = 1$ so the constraint (22) enforces $T_{ijg} = 1$. If $P_{ijg} = 1$, $Q_{ijg} = 0$, then constraint (25) becomes void and constraint (24) enforces $e_{ijg} \leq T_{ijg}$.
- If $Q_{ijg} = x_{ig}x_{jg} = 0$ and $P_{ijg} = 0$, then constraints (22)–(23) cannot enforce any value on T_{ijg} . If $P_{ijg} = 0$, $Q_{ijg} = 0$, then (24)–(25) enforce $e_{ijg} = T_{ijg}$.

Lastly, $e_{ijg} = e_{jig}, \forall i_g, j_g, \forall g \in \{2, \dots, t\}$ enforce the edges to be undirected.

4 Results

4.1 Synthetic Data

Our method ILP-DMR is evaluated with respect to the greedy method ReverseDMR [14] over simulations with known evolutionary histories. We simulate DMR graph evolution by starting on a graph with two vertices with an edge between them. In each case, we provide a present-day graph to ILP-DMR and ReverseDMR as input. We limit the maximum running time of ILP-DMR to a single day for every inference. We use Gurobi [5] to implement ILP-DMR.

We have evaluated the history reconstruction performance in terms of 3 metrics. The first metric is the reconstructed entire history’s likelihood. Secondly, evaluating the vertex arrival order as part of evolution is another metric. Vertex arrival order for an estimated history is calculated via reversing the set of vertices removed at each step. Then, the relation between the true evolution’s arrival order and that reversed list of nodes is estimated by Kendall’s Tau [8]. There are a number of Kendall’s Tau versions and we account for ties in our evaluation case. Let Q be set of discordant pairs, P be the set of concordant pairs, U be the set of ties only in A_R , and T be the set of ties only in A_T . Then, the score is given as $\tau = \frac{|P| - |Q|}{\sqrt{(|P| + |Q| + |T|) * (|P| + |Q| + |U|)}}$. We do not add a

tie to either U or T if it exists for the identical pair in both A_R and A_T . More formally, the considered observation pairs $(x_i, y_i), (x_j, y_j)$ satisfy the following: 1- $x_i, x_j \in A_T, y_i, y_j \in A_R$, and 2- $i < j$. Concordant pair $(x_i, y_i), (x_j, y_j)$ is defined as a pair where both items ranks are in agreement, such as both $x_i < x_j$ and $y_i < y_j$; or both $x_i > x_j$ and $y_i > y_j$. Discordant pair $(x_i, y_i), (x_j, y_j)$ is defined as a pair where $x_i > x_j$ and $y_i < y_j$ or $x_i < x_j$ and $y_i > y_j$. Otherwise, a pair is considered a tie if $x_i = x_j$ or $y_i = y_j$. A higher value means better performance for both Kendall’s Tau and likelihood metrics.

Thirdly, we compare the graphs evolved at each reconstruction step with the corresponding correct graphs by graph kernel [21]. For a present graph at time t G_t , we are provided with the reconstructed snapshots of the graph $\hat{G}_S = \hat{G}_3, \dots, \hat{G}_{t-1}$ as well as true graphs at previous time steps $G_S = G_3, \dots, G_{t-1}$. Then, kernel similarity is calculated as $\sum_{i=3}^{t-1} k(G_i, \hat{G}_i)/(t-2)$ where k defines a graph kernel. That kernel similarity-based score is a measure of the whole reconstruction’s mean similarity. In our case, the Weisfeiler-Lehman kernel is used which depends on subgraph isomorphism inside the input networks [18]. A higher value means better performance for kernel similarity.

Reconstruction with True Model Parameters. We have generated 1500 synthetic graphs where extant network’s vertex count varies from 6 to 10. For each graph, we choose DMR parameters q_{del} and q_{mod} randomly in $[0.1, 0.9]$ ranges, by rounding them to a single decimal. Then, we use the identical DMR parameters while reconstructing the history for ILP-DMR and ReverseDMR.

Figure 1 shows that there were no simulations where solutions from ILP have a lower likelihood than that of ReverseDMR. Since these are small networks, both ReverseDMR and ILP were able to find optimal solutions in 76% of the cases, while in 24% of the cases ILP found solutions with higher likelihood. The fact that ILP could find solutions with higher likelihood shows that ReverseDMR is not guaranteed to find optimal solutions. Histories reconstructed from ILP had better correlation with the true histories, with respect to node arrival order, in 93% of the cases. The kernel similarity values of the reconstructed networks were not lower than those from ReverseDMR in 79% of the cases.

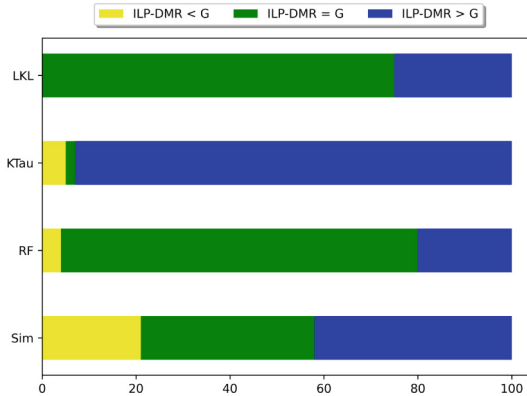


Fig. 1. Reconstruction with true model parameters. Proportion of simulations where reconstructed histories from ILP scored better ($ILP > G$), equal ($ILP=G$) and worse ($ILP < G$) than the reconstructions from ReverseDMR, the Greedy approach [14], for three metrics: 1- Log-likelihood (LKL), 2- Kendall’s Tau (KTAU), and 3- Kernel Similarity (Sim).

Overall, reconstructed histories from ILP have higher likelihood and obtain node arrival orders, duplication events and inferred networks that are closer to the true evolutionary history compared to those from ReverseDMR.

4.2 Real Networks

We have evaluated the performance of ILP-DMR and ReverseDMR over real dataset by reconstructing bZIP transcription factors PPI network histories. We executed each of these algorithms over q_{del} and q_{mod} values in ranges $[0.1, 0.3, 0.7, 0.9]$ and $[0.4, 0.7]$ respectively. Then, we choose the solution with the highest log-likelihood to analyze in depth. We limit the maximum execution time of ILP-DMR to a single day for each execution. We initialize ILP solver by using ReverseDMR's feasible solution. Among the solutions generated by the solver, we examine 2 solutions (ILP-DMR1 and ILP-DMR2) for every extant graph.

In real network setting, we could not evaluate the performance by the 3 metrics utilized in the previous synthetic network section, as we do not know the correct histories. In this case, we focus on evaluating the results from a biological perspective in 2 ways. First of all, we follow the process mentioned in [14] to compare the history inferences vertex arrival times. We estimate protein arrival times approximately via existing ortholog knowledge, where we assume that early-arriving proteins tend to have more ortholog proteins. So, we sort the set of proteins in the present-day network by decreasing number of ortholog proteins they have, which is then assumed to be the 'correct' vertex arrival order (AT). The number of ortholog proteins for every protein is determined by querying OrthoDB [9]. In OrthoDB, we count the gene frequency at the highest level at which ortholog knowledge is available for the whole set of proteins in the graphs. Reconstructed histories from both ReverseDMR and ILP-DMR capture the eliminated vertex at each time step, which defines the inferred vertex arrival order (AR) for each of these methods. Then, we compare two ranked lists AR and AT via Kendall's Tau [8] which finds the correlation between them.

We also evaluate the similarities between sequences of estimated anchors and duplicated vertices. In DMR models each evolutionary step, anchor gene (a) is copied to a new gene (d), so the similarity between a and d is expected to be greater than the similarity between anchor gene and other genes at each step.

Provided with the present-day graph G_t , this graph's inference history $\hat{G}_S = \hat{G}_3, \dots, \hat{G}_{t-1}, G_t$, as well as selected anchor vertices and duplicated vertices in each graph, $\rho(\hat{G}_i)$ score is computed for each graph in $\hat{G}_i \in \hat{G}_S$ by sequence similarity between the selected anchor and duplicated vertex protein pairs. The ultimate inference metric is called Anchor Duplicate Similarity Score (ADSS), and it is calculated as $\sum_{\hat{G}_i \in \hat{G}_S} \rho(\hat{G}_i) / (t - 2)$ which is normalized via total graph count in \hat{G}_S . Here, we do not consider \hat{G}_2 as only a single gene exists as part of the initial duplication from \hat{G}_1 to \hat{G}_2 . So, no remaining genes exist to be compared. When provided with 2 inference of the identical present-day graph, the better anchor and duplicated vertices are chosen, the higher ADSS becomes.

bZIP Transcription Factors. Protein family basic-region leucine zipper (bZIP) consists of transcription factors, and it is crucial for a number of tasks such as circadian clock, stress response, etc. [1, 16]. These proteins coiled leucine zipper domains are quite important for mediating the interactions among those proteins. Therefore, only sequence content of these proteins may predict those interactions quality more precisely [4]. [16] inferred present-day interaction graphs between a list of bZIP proteins across different species, by using this sequence similarity concept in [4]. In our case, we focus on Homo sapiens graph and obtained the graph by merging the same protein’s subunits into a single vertex.

In terms of inferring the ancestry of this bZIP graph, we focus on 3 metrics for performance evaluation as seen in Table 1: Vertex Arrival Time, Accuracy by Kendall’s Tau, ADSS, and likelihood of ReverseDMR and ILP-DMR solutions. In terms of all these criteria, ReverseDMR performs worse than the 2 solutions from ILP-DMR (ILP-DMR1 and ILP-DMR2). In this case, ILP2 performs better in terms of ADSS, whereas ILP-DMR1 performs better in terms of Kendall’s Tau. ReverseDMR’s and these 2 ILP-DMR solutions order of protein arrivals are shown in Table 2. The leftmost column shows the true order estimated from ortholog knowledge which is used in Kendall’s Tau calculation.

Table 1. For DMR parameters $q_{\text{del}} = 0.3$, $q_{\text{mod}} = 0.4$ over reconstructing the ancestry of bZIP graph, ADSS, vertex arrival accuracy, and likelihood are shown.

Algorithm	Log-likelihood	Kendall’s Tau	ADSS
ReverseDMR	-21	-0.23	-1901.55
ILP-DMR1	-19.6	0.18	-1797.11
ILP-DMR2	-19.6	-0.03	-1749.31

Table 2. The left column shows bZIP network’s anchor proteins in number of orthologs decreasing order, whereas the rightmost columns show the appearance order at each step over ReverseDMR and 2 ILP-DMR solutions inferences.

Gene	Orthologs	Time step	Reverse DMR	ILP-DMR1	ILP-DMR2
JUN	820	2	FOS,	ATF6,	FOS,
ATF2	546		CREB	ATF2	ATF2
ATF6	453	3	ATF6	FOS	XBP1
ATF4	432	4	BATF	ATF3	ATF3
FOS	339	5	ATF4	CREB	JUN
OASIS	285	6	ATF2	JUN	CREB
CREB	268	7	E4BP4	CEBP	CEBP
CEBP	259	8	JUN	BATF	BATF
E4BP4	255	9	ATF3	PAR	PAR
PAR	244	10	CEBP	ATF4	ATF4
ATF3	229	11	PAR	E4BP4	E4BP4
BATF	227	12	XBP1	OASIS	OASIS
XBP1	203	13	OASIS	XBP1	ATF6

Extra proof for our inference's correctness is provided by analyzing bZIP transcription factor sequences phylogenetically as in [1]. Such analysis has identified a remarkably preserved core graph including ATF3, FOS, and JUN proteins. Those 3 proteins arrive earlier in both ILP-DMR1 and ILP-DMR2 (before the 7th step), whereas ATF3 and JUN appear in the solution after the 7th step when reconstructed via ReverseDMR as in Table 2. The main distinction between ILP-DMR1 and ILP-DMR2 solutions is the different arrival order for XBP1 and ATF6. Even though there exists an interaction between ATF6 and XBP1 in unfolded protein response pathway [13], to our best knowledge, there is no preference for earlier appearance of ATF6 over XBP1 as well as XBP1 over ATF6.

5 Conclusions

In this paper, we come up with ILP-DMR, an ILP-based solution, to reconstruct PPI network's evolution under DMR model by maximizing the likelihood. Since ILPs are generally NP-hard, reconstruction histories may not be optimally identified in polynomial time for very large networks. Nevertheless, we can use general-purpose ILP solver heuristics to infer a number of nearly optimal solutions in a predetermined amount of time. We evaluated the performance of our ILP-DMR solutions by comparing them with the baseline ReverseDMR's solutions, which was previously the best-performing method for the reconstruction. Over synthetic datasets, our ILP method outperforms ReverseDMR in terms of all 3 metrics; likelihood, vertex arrival order, and inferred ancestral graphs similarity to the correct graphs. Over real bZIP protein network, the likelihoods of ILP-DMR solutions were higher and ILP-DMR solutions agreed stronger with sequence similarity and ortholog knowledge. ILP-based graph history reconstruction can be seen as an important contribution for additional developments in graph reconstruction. For example, one can extend the parts of the proposed ILP-DMR to other graph growth models such as Forest Fire and Preferential Attachment model. We can generalize our ILP-based solution to integrate several input networks rather than one, and it can consider extra knowledge such as historical gene duplications.

References

1. Amoutzias, G., et al.: One billion years of bZIP transcription factor evolution: conservation and change in dimerization and DNA-binding site specificity. *Mol. Biol. Evol.* **24**(3), 827–835 (2006)
2. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999)
3. Colak, R., et al.: Dense graphlet statistics of protein interaction and random networks. *Pac. Symp. Biocomput.* **2009**, 178–189 (2009)
4. Fong, J.H., Keating, A.E., Singh, M.: Predicting specificity in bZIP coiled-coil protein interactions. *Genome Biol.* **5**(2), R11 (2004)

5. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2023). <https://www.gurobi.com>
6. Jasra, A., Persing, A., Beskos, A., Heine, K., De Iorio, M.: Bayesian inference for duplication-mutation with complementarity network models. *J. Comput. Biol.* **22**(11), 1025–1033 (2015)
7. Jin, Y., Turaev, D., Weinmaier, T., Rattei, T., Makse, H.A.: The evolutionary dynamics of protein-protein interaction networks inferred from the reconstruction of ancient networks. *PLoS ONE* **8**(3), 1–15 (2013)
8. Kendall, M.G.: The treatment of ties in ranking problems. *Biometrika* **33**(3), 239–251 (1945)
9. Kriventseva, E.V., et al.: OrthoDB v10: sampling the diversity of animal, plant, fungal, protist, bacterial and viral genomes for evolutionary and functional annotations of orthologs. *Nucleic Acids Res.* **47**(D1), D807–D811 (2018)
10. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over time: densification laws, shrinking diameters and possible explanations. In: *KDD 2005*, pp. 177–187. Association for Computing Machinery, New York (2005)
11. Li, S., Choi, K.P., Wu, T., Zhang, L.: Maximum likelihood inference of the evolutionary history of a PPI network from the duplication history of its proteins. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **10**(6), 1412–1421 (2013)
12. Middendorff, M., Ziv, E., Wiggins, C.H.: Inferring network mechanisms: the drosophila melanogaster protein interaction network. *Proc. Natl. Acad. Sci.* **102**(9), 3192–3197 (2005)
13. Mitra, S., Ryoo, H.D.: The unfolded protein response in metazoan development. *J. Cell Sci.* **132**(5), jcs217216 (2019)
14. Navlakha, S., Kingsford, C.: Network archaeology: uncovering ancient networks from present-day interactions. *PLoS Comput. Biol.* **7**(4), 1–16 (2011)
15. Patro, R., Sefer, E., Malin, J., Marçais, G., Navlakha, S., Kingsford, C.: Parsimonious reconstruction of network evolution. *Algorithms Mol. Biol.* **7**(1), 25 (2012)
16. Pinney, J.W., Amoutzias, G.D., Rattray, M., Robertson, D.L.: Reconstruction of ancestral protein interaction networks for the bZIP transcription factors. *Proc. Natl. Acad. Sci.* **104**(51), 20449–20453 (2007)
17. Rajan, V., Zhang, Z., Kingsford, C., Zhang, X.: Maximum likelihood reconstruction of ancestral networks by integer linear programming. *Bioinformatics* **37**(8), 1083–1092 (2020)
18. Shervashidze, N., Schweitzer, P., van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-lehman graph kernels. *J. Mach. Learn. Res.* **12**(77), 2539–2561 (2011)
19. Sreedharan, J.K., Magner, A., Grama, A., Szpankowski, W.: Inferring temporal information from a snapshot of a dynamic network. *Sci. Rep.* **9**(1), 3057 (2019)
20. Vázquez, A., Flammini, A., Maritan, A., Vespignani, A.: Modeling of protein interaction networks. *Complexus* **1**(1), 38–44 (2003)
21. Vishwanathan, S.V.N., Schraudolph, N.N., Kondor, R., Borgwardt, K.M.: Graph kernels. *J. Mach. Learn. Res.* **11**, 1201–1242 (2010)
22. Wagner, A.: The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. *Mol. Biol. Evol.* **18**(7), 1283–1292 (2001)
23. Young, J.G., St-Onge, G., Laurence, E., Murphy, C., Hébert-Dufresne, L., Desrosiers, P.: Phase transition in the recoverability of network history. *Phys. Rev. X* **9**, 041056 (2019)



Farthest-First Traversal for Identifying Multiple Influential Spreaders

Madhvi Ramrakhiyani^(✉), Mukesh Tiwari, and V. Sunitha

Dhirubhai Ambani Institute of Information and Communication Technology
(DA-IICT), Gandhinagar, Gujarat, India
{madhvi_ramrakhiyani,mukesh_tiwari,v_suni}@daiict.ac.in

Abstract. We propose a method for identifying multiple influential spreaders in complex networks. This method is based on a farthest-first traversal of the network. The spreaders selected by this method satisfy the two criteria of being dispersed as well as influential in their neighborhood. To examine the influence of the spreaders identified by our method, we perform numerical simulations of SIR-based epidemic spread dynamics. For a range of parameter values, we observe that the epidemic size obtained using the spreaders generated by our method as the initial spreaders is at least as large as the epidemic size obtained using hubs as initial spreaders.

Keywords: Influential spreaders · Diffusion on networks · Farthest-first traversal

1 Introduction

Understanding the dynamics of spread processes in social and biological systems is a challenging problem. Significant insight, however, into these processes has emerged due to the advances in network science over the past two decades [1, 4, 7, 24]. Modeling the spread as a process on complex networks has allowed us to discern the different aspects of the influence of the complex network properties on the dynamics of spread [18].

In many situations, such as the explosive spread of rumors and false information [22], rapid diffusion of contagion in a population [26], targeted immunization [19], network resilience [8], robustness and failure of power grid and transportation networks [9], identifying opinion leaders for rapid behavior change [23], identifying the node(s) central to the process in the network representation becomes essential. A node is considered central or influential if it can affect many other nodes in the network. Identifying such nodes, however, is non-trivial since it depends on the heterogeneity of the network structure. In addition, the definition of an influential node also depends on the problem under study. Various ranking methods based on the network's standard structural measures such as degree, closeness, betweenness, have been extensively studied [5, 17]. In addition to the classical centrality that are based on topological properties, centralities

with high influence at the meso level such as through communities have also been explored [11, 29]. However, a general universal framework for identifying the most influential node does not apply to all problems [6, 16].

Apart from studies aimed at analyzing the impact of a single influential spreader, recent research has also focussed on the influence maximization problem [14]. In this case, the objective is to identify a set of nodes that result in maximum expected influence spread. If the spread process starts at multiple locations, then selecting nodes based on ranking alone does not result in an optimal spread. Kitsak *et al.* [15] highlighted the role of position of influential node by identifying them based on k-shell decomposition. In addition, they also underscored the importance of separation between high-degree nodes when considering multiple influential spreaders. Subsequently, the effect of the separation between initial spreaders on spread efficiency was analyzed on regular and small world lattices [13]. Graph burning [3] has been used as a model for the spread of influence in social networks. Since the graph burning problem is NP-hard, any set of influential spreaders obtained using this model can only be approximate. Here, we would like to point out that a farthest-first traversal based approximation for the burning sequence of a network has been proposed in [10].

Selecting a set of nodes that are influential as well as scattered is not straightforward. Ranking schemes such as graph coloring based method [28], VoteRank [27], heuristic clustering algorithm based [2], selecting influential spreaders with low overlapping influences [30], partition and community-based methods [12, 25] aim to identify initial spreaders that satisfy the two criteria; viz., of multiple spreaders being dispersed as well as influential in their neighborhood. These methods observe a more extensive spread compared to the classical network centrality-based methods.

In this paper, we propose a simple distance-based method to identify a set of influential spreaders in the network. In the method, first, a set of nodes is selected through a farthest-first (FF) traversal. The selected nodes are used to partition the network into non-overlapping clusters. For each cluster, the node with the maximum number of links within the cluster is assumed to be the most influential. These nodes are used to initiate a spread process on the network. In this paper, we report our observations on the epidemic size (using the SIR epidemic model) through the influential spreaders selected by our proposed method. We also compare our observations with those obtained from initial spreaders chosen based on the degree centrality ranks and the FF nodes as initial spreaders for different data sets.

The rest of the paper is laid out as follows: In Sect. 2, we discuss the details of our proposed method. In Sect. 3 we present the results of the spread obtained through numerical experiments of SIR model on different network based datasets. Conclusions are presented at the end.

2 Method

The greedy method for the FF traversal was introduced by Rosenkrantz et al. [20] as a solution for the traveling salesman problem. The FF traversal is a sequential process of node selection in a graph. It starts with an arbitrarily chosen seed node. At each step, the next node is determined as the one farthest from all the previously selected nodes. Selection of a certain number of FF nodes in this manner is the pre-processing stage of our method. It must be noted that the set of FF nodes selected is dependent on the seed node used for the FF traversal. Partitioning the network using the selected FF nodes is the next stage of our proposed method. The final stage of our method is to identify a node from each part of this partition as a candidate for the set of influential spreaders. While the selected FF nodes ensure separation between them, these nodes are not necessarily influential. However, FF nodes can be used to perform a search for the nodes with a high local influence. The local search radius is, however, arbitrary. In this paper, we partition the network into clusters with the FF nodes at the center of each cluster. Each node belongs to the partition of the FF node to which it is closest. If the shortest path of a node is equidistant to multiple FF nodes, then it randomly selects a nearest cluster.

The node with the largest intra-partition degree in each cluster is selected as the most influential spreader in that cluster. This guarantees that the node is able to influence a large number of nodes within its cluster. It should be noted that while separation between FF nodes is ensured, it does not necessarily apply to the set of influential spreaders. Separation is more likely if influential spreaders are selected from the set of nearest neighbors of the FF nodes. However, in our numerical experiments we have observed the influential spreaders to have reasonable separation.

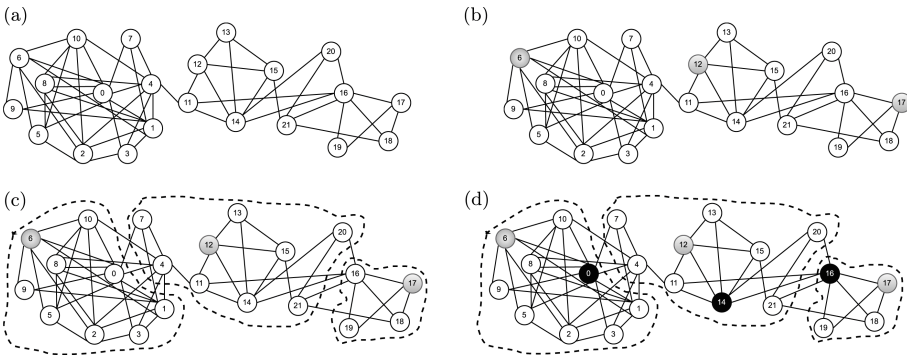


Fig. 1. (a) A toy network. In (b) we show a realization of the selection of FF nodes (shaded grey) and in (c) the partition created by the FF nodes. The different parts of the partition are demarcated by dashed lines. In (d) the hub in the sub-graph induced by the vertices of a part is selected as the influential node (shaded black) for that part.

In Fig. 1, we demonstrate the proposed method through a toy network (Fig. 1(a)). In Fig. 1(b), three FF nodes, (6, 12 and 17) are shown, starting with the node labeled 6 as the seed node. Subsequently, as shown in Fig. 1(c), the network is partitioned into clusters. Since a node randomly chooses a cluster if it is equidistant from multiple FF nodes, the cluster radii are different. Node labeled 4 is equidistant from FF nodes 6 and 12 and in the instance presented here, chooses to join the cluster represented by node 12. In Fig. 1(d), the node with the largest intra-degree in a cluster is selected as the influential node in that cluster. As stated earlier, the set of nodes obtained by combining the influential nodes from each cluster forms the most influential spreaders in our proposed method (nodes 0, 14, and 16 here). It should be noted that even though nodes 1 and 4 have a higher degree they do not have a strong intra-partition influence and therefore are not present in the list of influential spreaders.

3 Results

We explore the spread efficiency of the influential spreaders obtained by the proposed method on different network data sets. We have primarily focussed on the data sets that differ in diameter. Four data sets, of which three are spatial, namely, minnesota, euroroads, and US powergrid, and one non-spatial, GRQC collaboration network, have been considered here. In Table 1, we list their details. These include the total number of nodes (N), total number of edges (E), density (ρ), average clustering coefficient (C), assortativity coefficient (AC), diameter (D), radius (R) and average path length ($\langle l \rangle$).

To study the influence of the initial spreaders, we use the susceptible-infected-recovered (SIR) model of epidemic spread. Relevant to the SIR model, we use β , which denotes the probability that an infected node will infect its neighbor, and γ , which represents an infected node's recovery probability. We consider small values of β and γ that allow comparison of the influence of spreaders obtained by different methods. For diffusion processes modeled as the epidemic spread, the size of the epidemic is defined as the value of the recovered in the long time limit $R(t \rightarrow \infty)$. The epidemic spread is initiated from an initial set of influential spreaders. For the results presented here, 1% of the nodes are initially assumed to be infected. To highlight the role of distance and influence, we compare the influence size from the proposed method to that obtained from FF nodes and

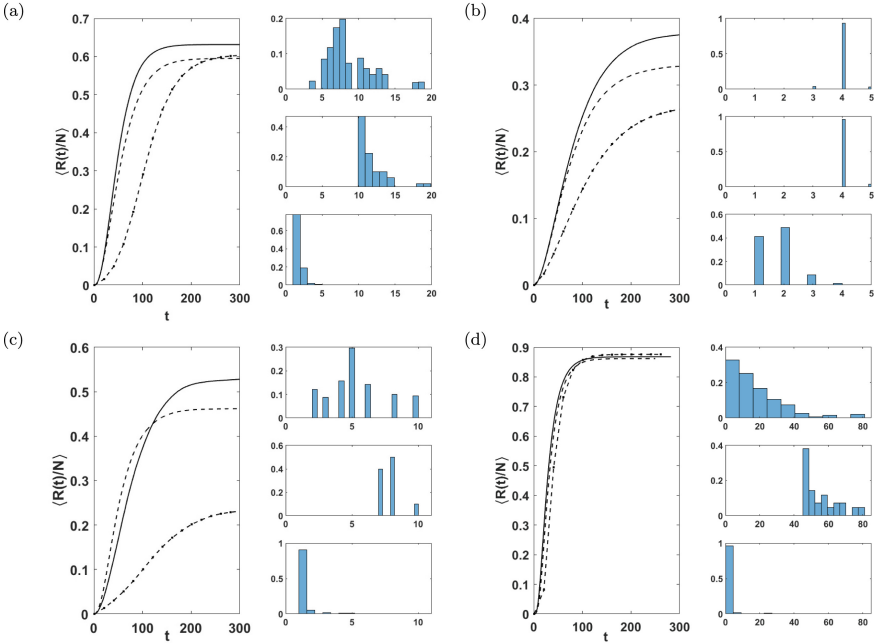


Fig. 2. Comparison of the average fraction of recovered ($\langle R(t)/N \rangle$) for SIR simulation on (a) US powergrid, (b) minnesota, (c) euroroads, and (d) GRQC data sets. The solid line shows the spread from FF traversal-based influential spreaders selection, the dashed line from influential spreaders selected based on degree centrality, and the dashed-dotted line from the FF nodes. In the right panel of each figure, the distribution of the degree of the initial spreaders selected based on (top) FF traversal-based influential spreader (middle) degree rank and (bottom) FF nodes is shown. The number of samples taken is 100, and values of β and γ are 0.1 and 0.05 respectively.

Table 1. Details of network datasets used in the numerical experiments and their structural properties. The datasets were obtained from [21].

Data set	N	E	ρ	C	AC	D	R	$\langle l \rangle$
minnesota	2640	3302	0.00094	0.015	-0.18	99	52	35.34
euroroads	1039	1305	0.0024	0.0189	0.0900	62	31	18.3
US powergrid	4941	6594	0.0005	0.0801	0.0034	46	23	18.98
GRQC	4158	13428	0.0015	0.5568	0.63	17	9	6.04

nodes selected based on their degree centrality rank for the same parameter values.

In Fig. 2, we show the evolution of the fraction of recovered averaged over 100 samples for the four data sets. Except for the GRQC data set, influential spreaders selected through our proposed method are observed to result in a larger

epidemic size. The size of the spread initiated from FF nodes, on the other hand, is always the smallest. Separation alone is, therefore, not sufficient for a larger influence. In Fig. 2 we have also shown the distribution of the degree of initial spreaders selected by the three methods for each data set. The degrees of FF nodes are always smaller compared to the other two methods. On the other hand, initial spreaders selected through the proposed method have a broader range of degrees, whose average, except for the minnesota data set, is smaller than those of high-degree nodes.

Figure 3, in which we mark the set of initial spreaders selected by the different methods for a single sample, provides further insight into our observations. The set of FF nodes (marked in black) are located in the distant parts of the network. This also explains the low degrees of these nodes since their location is mostly on the periphery of networks. Influential spreaders selected through the proposed method (marked in red) are scattered throughout the network. Therefore, both FF and FF traversal-based influential nodes provide wider coverage on the network, reaching out to the sparsely connected regions. Coverage of degree-ranked initial spreaders (marked in green), on the other hand, is less. In the minnesota data set, the variation in degrees of different nodes is not large. Nevertheless, a larger epidemic size for initial spreaders selected from the proposed method highlights the importance of separation between initial influential spreaders.

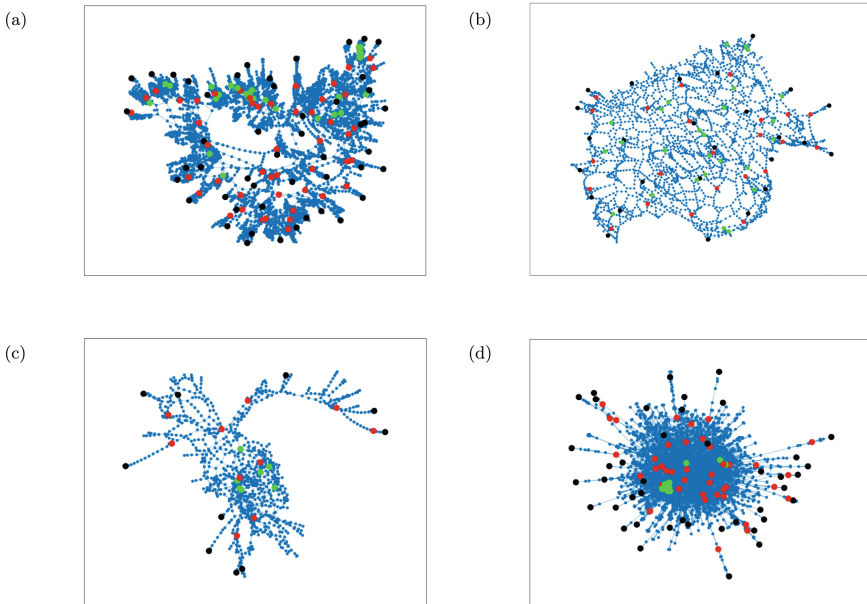


Fig. 3. Node selection in a single sample for (a) US powergrid, (b) minnesota, (c) euroroads, and (d) GRQC data set. FF nodes are marked in black, FF traversal-based initial spreaders in red, and degree-ranked nodes in green.

4 Conclusions

This paper proposes an FF traversal-based method to identify influential spreaders in a complex network. Our proposed method has three steps. In the first step, a list of nodes separated on the network is generated using the FF traversal. In the second step, the network is partitioned into clusters with the FF nodes as the nearby nodes' attractors. In the final step, the node with the largest intra-degree in a cluster is added to the list of influential spreaders. By selecting an influential node per cluster, we have studied the epidemic size on different network data sets. Our numerical experiments show that the influence of nodes selected by the proposed method can be larger than that of nodes selected through degree centrality in the case of spatial networks. The separation between influential spreaders will have a larger effect if the diameter is large. In contrast, it may be insignificant if the dynamics are governed by hub nodes, as in scale-free networks. The proposed method allows the selection of influential spreaders in distant parts of the network. It should help expand the coverage in large spatial networks.

References

1. Aral, S., Walker, D.: Identifying influential and susceptible members of social networks. *Science* **337**(6092), 337–341 (2012)
2. Bao, Z.K., Liu, J.G., Zhang, H.F.: Identifying multiple influential spreaders by a heuristic clustering algorithm. *Phys. Lett. A* **381**(11), 976–983 (2017)
3. Bonato, A., Janssen, J., Roshanbin, E.: Burning a graph as a model of social contagion. In: Bonato, A., Graham, F.C., Pralat, P. (eds.) WAW 2014. LNCS, vol. 8882, pp. 13–22. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13123-8_2
4. Centola, D.: The spread of behavior in an online social network experiment. *Science* **329**(5996), 1194–1197 (2010)
5. Da Silva, R.A.P., Viana, M.P., da Fontoura Costa, L.: Predicting epidemic outbreak from individual features of the spreaders. *J. Stat. Mech. Theory Exp.* **2012**(07), P07005 (2012)
6. De Arruda, G.F., Barbieri, A.L., Rodríguez, P.M., Rodrigues, F.A., Moreno, Y., da Fontoura Costa, L.: Role of centrality for the identification of influential spreaders in complex networks. *Phys. Rev. E* **90**(3), 032812 (2014)
7. De Arruda, G.F., Rodrigues, F.A., Moreno, Y.: Fundamentals of spreading processes in single and multilayer complex networks. *Phys. Rep.* **756**, 1–59 (2018)
8. Gao, J., Barzel, B., Barabási, A.L.: Universal resilience patterns in complex networks. *Nature* **530**(7590), 307–312 (2016)
9. Gao, J., Buldyrev, S.V., Havlin, S., Stanley, H.E.: Robustness of a network of networks. *Phys. Rev. Lett.* **107**(19), 195701 (2011)
10. García-Díaz, J., Pérez-Sansalvador, J.C., Rodríguez-Henríquez, L.M.X., Cornejo-Acosta, J.A.: Burning graphs through farthest-first traversal. *IEEE Access* **10**, 30395–30404 (2022)
11. Ghalmane, Z., El Hassouni, M., Cherifi, H.: Immunization of networks with non-overlapping community structure. *Soc. Netw. Anal. Min.* **9**(1), 1–22 (2019)
12. He, J.L., Fu, Y., Chen, D.B.: A novel top-k strategy for influence maximization in complex networks with community structure. *PloS one* **10**(12), e0145283 (2015)

13. Hu, Z.L., Liu, J.G., Yang, G.Y., Ren, Z.M.: Effects of the distance among multiple spreaders on the spreading. *Europhys. Lett.* **106**(1), 18002 (2014)
14. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 137–146 (2003)
15. Kitsak, M., et al.: Identification of influential spreaders in complex networks. *Nat. Phys.* **6**(11), 888–893 (2010)
16. Lalou, M., Tahraoui, M.A., Kheddouci, H.: The critical node detection problem in networks: a survey. *Comput. Sci. Rev.* **28**, 92–117 (2018)
17. Lü, L., Chen, D., Ren, X.L., Zhang, Q.M., Zhang, Y.C., Zhou, T.: Vital nodes identification in complex networks. *Phys. Rep.* **650**, 1–63 (2016)
18. Pastor-Satorras, R., Castellano, C., Van Mieghem, P., Vespignani, A.: Epidemic processes in complex networks. *Rev. Mod. Phys.* **87**(3), 925 (2015)
19. Pastor-Satorras, R., Vespignani, A.: Immunization of complex networks. *Phys. Rev. E* **65**(3), 036104 (2002)
20. Rosenkrantz, D.J., Stearns, R.E., Lewis, P.M., II.: An analysis of several heuristics for the traveling salesman problem. *SIAM J. Comput.* **6**(3), 563–581 (1977)
21. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: *AAAI* (2015). <https://networkrepository.com>
22. Shao, C., Ciampaglia, G.L., Varol, O., Yang, K.C., Flammini, A., Menczer, F.: The spread of low-credibility content by social bots. *Nat. Commun.* **9**(1), 1–9 (2018)
23. Valente, T.W., Pumpuang, P.: Identifying opinion leaders to promote behavior change. *Health Educ. Behav.* **34**(6), 881–896 (2007)
24. Wang, W., Liu, Q.H., Liang, J., Hu, Y., Zhou, T.: Coevolution spreading in complex networks. *Phys. Rep.* **820**, 1–51 (2019)
25. Yanez-Sierra, J., Diaz-Perez, A., Sosa-Sosa, V.: An efficient partition-based approach to identify and scatter multiple relevant spreaders in complex networks. *Entropy* **23**(9), 1216 (2021)
26. Zhang, D., Wang, Y., Zhang, Z.: Identifying and quantifying potential super-spreaders in social networks. *Sci. Rep.* **9**(1), 14811 (2019)
27. Zhang, J.X., Chen, D.B., Dong, Q., Zhao, Z.D.: Identifying a set of influential spreaders in complex networks. *Sci. Rep.* **6**(1), 27823 (2016)
28. Zhao, X.Y., Huang, B., Tang, M., Zhang, H.F., Chen, D.B.: Identifying effective multiple spreaders by coloring complex networks. *Europhys. Lett.* **108**(6), 68005 (2015)
29. Zhao, Z., Wang, X., Zhang, W., Zhu, Z.: A community-based approach to identifying influential spreaders. *Entropy* **17**(4), 2228–2252 (2015)
30. Zhou, M.Y., Xiong, W.M., Wu, X.Y., Zhang, Y.X., Liao, H.: Overlapping influence inspires the selection of multiple spreaders in complex networks. *Phys. A* **508**, 76–83 (2018)



Wishful Thinking About Consciousness

Peter Grindrod^(✉)

Mathematical Institute, University of Oxford, Oxford, UK
grindrod@maths.ox.ac.uk

Abstract. We contrast three distinct mathematical approaches to the hard problem of consciousness: quantum consciousness, integrated information theory, and the very large scale dynamical systems simulation of a network of networks. We highlight their features and their associated hypotheses, and we discuss how they are aligned or in conflict. We suggest some challenges for the future theories, in considering how they might apply to the human brain as it develops both cognitive and conscious sophistication, from infancy to adulthood; and how an evolutionary perspective challenges the distinct approaches to aver performance advantages and physiological surrogates for consciousness.

Keywords: Quantum consciousness · Integrated information theory · Dynamical systems simulation · evolutionary advantage · performance surrogates

1 Introduction

The *hard problem of consciousness* [1, 2] asks why there is something internal to our subjective experience, some set of phenomenological sensations, something that it is *like* to be a human brain experiencing the world. Such repeatable and consistent sensations range from large scale emotions and feelings (anxiety, happiness, love, embarrassment) down to smaller scale, more specific, *qualia* (headache pain, the sight of the blueness of blue, the brassy sound of a trumpet, the feel of stroking cat's fur, the crunch from biting into an apple, ...). These are internal mental states with very distinctive subjective characters. How do such sensations come about within the physical brain and what is their possible role?

In this paper we consider three alternative mathematical approaches to the hard problem and related matters. We do so in order to crystallize and contrast the pros and cons of each paradigm. We wish to avoid a *dialogue of the deaf*. We hope that a direct inter-comparison will stimulate interests and research within all three theories. Comparison, cross fertilization and competition are huge drivers within science and the present stage of the *mathematics of consciousness* demands a sharpening of its aims and objectives.

We should stress that other approaches are available, so we have focused here upon the three that are arguably most dominant at the present time.

1.1 Quantum Consciousness (QC)

There are many scientists and writers who hold that conscious phenomena, such as the existence, causes, and role, of internal phenomenological sensations (emotions, feelings, and qualia) relate to some type of quantum effects taking place somewhere within the physical brain, usually associated with the cognitive processing of information to produce consequent inferences and actions. Penrose proposed a type of wave function collapse, called *objective reduction*, from which consciousness phenomena are born [3].

Quantum consciousness (QC) usually starts out from a negative: that classical mathematics (dynamical systems, and other concepts) alone cannot explain consciousness, positing instead that quantum-mechanical phenomena, such as entanglement and superposition, might play an important part in the brain's function and could explain critical aspects of consciousness. Up until a few years ago perhaps the best evidence for this assertion was indeed the failure of those classical mathematical methods to define and substantiate a model that might expose the “how, what, why” of conscious phenomena. This is no longer the case. In Sects. 1.2 and 1.3, below, we will discuss two now obvious, and available, alternative candidates: integrated information theory (IIT), and the reverse engineering of very large scale (VLS) dynamical system simulations (DSS). The former required a novel concept to be applied to information processing systems; whilst the latter could not be prosecuted until large simulations on (multi-core) super-computing platforms became available, or else until a suitable simplification of the VLS systems could be defined.

So the time is ripe to reconsider the logic and evidence behind the promulgation of QC.

The quantum mind remains a hypothetical speculation, as Penrose and others admit. Until it can support a prediction that is testable by experimentation, the hypotheses is not based on any empirical evidence. Indeed, quite recently Jedlicka [4] says of quantum biology (a superset of the quantum mind-brain), “The recent rise of quantum biology as an emerging field at the border between quantum physics and the life sciences suggests that quantum events could play a non-trivial role also in neuronal cells. Direct experimental evidence for this is still missing...”. In [5] there is a useful summary of the contemporary evidence both for and against there being a functional role for quantum effects in a range of biological (and physiological) systems. The authors find no clear evidence one way or the other and they couch their conclusions in weak conditional terms, suggesting further experimentation is required.

However, just as any evidence to support the presence of quantum effects within the brain remains elusive, it is also hard to obtain positive evidence that rules them out. The major theoretical argument against the QC hypothesis is the assertion that any quantum states in the brain would lose coherency before they reached a scale where they could be useful [6, 7]. Typical brain reactions are on the order of milliseconds, trillions of times longer than sub-picosecond quantum timescales. Over many years though, there have been successive attempts to be

more explicit about where and how quantum effects might be present within the brain [8].

In [9], the authors consider the future of quantum biology as a whole and address QC explicitly. Given the objections above, on the basis of time-scale and space-scale discrepancy between quantum effects and neuronal dynamics, they conclude that any “*potential theory of quantum effects in biological neural networks would thus have to show how the macroscopic dynamics of biological neural nets can emerge from coherent dynamics on a much smaller scale.*”

With the present lack of any positive evidence for QC, despite many years of searching, and the existence of some coherent theoretical arguments to its contrary, why then does the quantum consciousness hypothesis persist? Perhaps the largest force driving its adoption is subjective: it comes from the desires and aspirations of quantum scientists themselves, to have their own physics become relevant to one of the most elusive frontiers in science. This goes far beyond Chalmers’ “minimisation of mysteries” jibe: it would act as a magnet and an employment-creation opportunity for quantum physicists.

Of course, the recent rise in quantum technologies (including quantum computing, quantum sensing and quantum communication) within novel synthetic applications, lavishly funded via many national programmes, performs a similar, though much more rational, purpose. Moreover, within those non-brain fields there is a focus on fabricating novel effects in the lab and beyond, rather than on unpicking and understanding a particular existing natural complex system, such as the human brain.

More recent ideas about consciousness introduce modifications of the quantum-mechanical Schrödinger equation and discuss wave function collapse. For example Chalmers and McQueen [10] and others [11] consider the evolution of quantum states within the universe when consciousness is also taken into account. They investigate whether conscious phenomena (within some paradigm) might collapse wave functions, inducing hard certainties. Such a role is normally reserved for acts of *observation* in quantum mechanics, though that is an ambiguous term. Hence they postulate that conscious phenomena (whether physical or dualist) could impact upon the real external world. Of course, this is the exact reverse of investigating whether or how quantum collapse might beget QC.

1.2 Integrated Information Theory (IIT)

Integrated information theory (IIT) [12] provides a framework capable of explaining why some physical systems (such as human brains) are conscious, why they feel the particular way they do in particular states, and whether other physical systems might be conscious. IIT does not build conscious-like phenomena out of physical systems and processes (as does dynamical systems modelling and simulation, discussed in Sect. 1.3 below), instead it moves from the abstract phenomenology towards mechanism by attempting to identify the properties of conscious experience within general information processing systems.

Here a *system* refers to a set of elements each of which might be in two or more discrete internal states. The state of the system is thus summarised by the

states of all of its elements. Subsets of the elements define “mechanisms”, and when the corresponding elements change state they do so in a way that may be conditional on one another’s state, since they inter-dependent and are able to interact. There is thus a transition matrix that can stipulate the probability that state of the system might switch to another state. IIT applies to whole systems that are capable of carrying out such internal dynamical state changes: it is an integrated view. In a real sense systems should be irreducible, since if they could be reduced (partitioned) into independent subsets then there would be no point in assembling those subsets into the whole and we might deal with each separately. This is akin to the notion of irreducibility (strong connectedness) for non-negative directed adjacency or dependency matrices (stipulating all pairwise influences between elements). Thus any properties of such an irreducible system are integrated and will depend upon all of its elements.

The details of IIT focus mainly on how a performance quantity called the “integrated information”, denoted by Φ , is defined and calculated for different systems. Φ is a real valued measure of the subsets of elements within a system that have (physical) cause-effect power upon one another. Only an irreducible (strongly connected) system full of feedback cycles can have a non-trivial Φ , as it produces output causes (consequences) from the incoming sensory effects. The conscious part of the human brain thus has a very high Φ , and is therefore highly conscious. Systems with a low Φ have a very small amount of consciousness.

In fact it is rather surprising how much effort is focused on the calculation of Φ , as a surrogate for the system’s internal *agility* and *sophistication*. This is apparent in the successively increasing formalism presented after a decade or so within IIT 3.0 in 2014 [13].

The mathematical essentials of IIT are well set out in [14], including its possible application within a quantum setting, introduced earlier in [15].

Of course, given any specific system, it would be *nice* to be able to calculate Φ , yet knowing its exact value is of no use to the system’s owner (except possibly for bragging). The owner continues to operate the system just as it is configured. Analogously we might all accept that there is a performance measure of human intelligence, called IQ, but knowing its actual value does not affect an individual’s own decision making or ability to operate as now. Of course a high value of Φ (like a high value of IQ) might confer some advantages to the system owner, such as having a comfortable life, or increased fecundity. It is easy to imagine how such advantages would cause some evolutionary selection to shift a population of owners to relatively higher and higher distributions of such measures. Thus the importance of higher Φ lies in its associated evolutionary advantages, not in its objective transparency or accessible calculation.

It is very interesting to ask how much improvement in Φ might be achieved if evolution re-architected the human brain; or even if individual (plastic) brains develop an abundance of connections when subject to specific training (specific experiences). Conversely, within a single operational lifetime, the brain’s consciousness development is not necessarily a one way street.

Equally, it is important to understand how Φ might increase as an infant brain develops through puberty, when both the cognitive sophistication and the conscious inner life develop along with the evolving neural connectivity and neurological structures, due to neurotransmitters and life experience.

Thus, the most important and appealing part of IIT is that it supplies a performance measure, Φ , as a system level attribute, that aims to be correlated with the level of internal conscious phenomena, and which might be increased. The ability to calculate Φ for any given class of systems is thus rather irrelevant to their owners – it is the internal consequences, that are measured by Φ , that will count. Any calculation of Φ is only relevant to demonstrating its well defined-ness and constructive nature, and possibly useful in future testing the IIT.

Like the quantum mind, IIT has its critics. The claims of IIT as a theory of consciousness are not yet scientifically established or testable [16], and IIT cannot be applied at the scale of a whole brain system. There is also no demonstration that systems which exhibit integration, in the sense of IIT, are in fact conscious at all. Obviously, a relatively high Φ -level might be a necessary condition for consciousness phenomena yet it may not be sufficient [17]. An explanatory gap remains.

1.3 Very Large Scale (VLS) Dynamical System Simulations (DSS)

Recent years have seen the possibility of VLS DSS containing 10B individual neurons, as a dynamic model for the human cortex. This approach is based on empirical observations of the cortex structure; it is an open system, subject to ongoing sensory inputs; it is experimental; and it is predictive. It makes predictions about why the cortex architecture should be so uniform (so to maximise the total dynamical degrees of freedom while constraining energy and volume) [18]; it explains how the whole system response is governed by (competing) internal dynamical modes [19] which result in a preconditioning of the immediate cognitive processing, providing a *fast thinking* advantage [20,21]; and it suggests that consciousness and cognition are entwined, with each catalysing and constraining the other, and the brain has evolved so as to exploit that advantage. Yet, as we shall see, there remains an explanatory gap [20].

In such VLS DSS neurons are arranged within a directed network architecture based on that of the human cortex. In fact, it is a *network-of-networks*. The inner networks, called *modules* (or communities) in network theory, each represent a single neural column containing 10,000 or so individual neurons which are internally very densely connected. The outer network connects up the neural columns with occasional connections between pairs of neurons from near-neighbouring columns. The columns are arranged in grid across the (flattened out) cerebral cortex. The individual neurons, just as in vivo, are both excitable (they spike when they stimulated by receiving an incoming spike) and refractory (following a spike they require a recovery time for the intra- and extra-cellular ions to re-equilibriate and they will not fire immediately if re-stimulated). Each directed neuron-to-neuron transmission takes some time, based on the tortuous nature of the individual axonal-synaptic-dendritic connection.

Recent work in such VLS DSS shows that under many distinct externally stimulated conditions the internal response defaults to react within one of a number of (hierarchically related) dynamical modes [19]. The modes exist across the cortex and across time and cannot be represented by snapshots, and are also mutually exclusive at any particular level in the hierarchy. Such VLS simulations require a supercomputer [22], and the reverse engineering of the internal responses to stimulation, and the identification of the hierarchically defined modes, is highly non-trivial [19].

The DSS approach recognises that the cognitive processing system in open, as it is constantly subject to sensory stimulation: it is not about dynamical *emergence* (symmetry breaking within disordered complex systems). The observed dynamical modes arise in response to various stimulations, and they are extremely good candidates for hierarchical emotions, feelings, and qualia. The hypothesis that internal phenomenological sensations correspond to the brain's own experiences of dynamical modes kicking-in directly addresses the *hard problem*: how humans have such internal sensations and exposing their role in enabling a fast thinking [21] evolutionary advantage by preconditioning immediate cognition and reducing the immediate decisions set.

Yet there remains an explanatory gap. While has been shown that any non-linear system of this type, including the human cortex, must have such internal competing dynamical modes, it has not be proven that these are in correspondence with internal phenomenological sensations. The set of internal modes is arranged hierarchically, and at any particular level they are mutually exclusive.

VLS DSS represents some of the largest numbers of simulations using massive cortex-like complex systems that have ever been made [23,24]. This endeavour requires significant resources. IBM has been particularly active and has carried out TrueNorth simulations in 2019 [25], realizing the vision of the 2008 DARPA Systems of Neuromorphic Adaptive Plastic Scalable Electronics (SyNAPSE) program. The simulations and analytics in [19] were carried out on the SpiNNaker 1 million-core platform [22,26,27].

The tribulations of two large science projects aiming to fully simulate human brains, within the US and EU, have been well documented [31]; and were caused by a variety of issues. These programmes have become focused on goals of brain mapping and building data processing facilities and new tools with which to study the brain. Many efforts have benefited from the computing facilities developed. The progress in [19], discussed above, exploited the massively parallel SpiNNaker supercomputer [22,26,27] that took over 10 years in construction, from 2006, and required £15M, funded by the UKRI/EPSCRC and the EU Human Brain Project [30].

In [31] these big science projects were summarised, “...*instead of answering the question of consciousness, developing these methods has, if anything, only opened up more questions about the brain-and shown just how complex it is.*”

In more recent work the modules (the neural columns) have been replaced by multi-dimensional clocks [32] (with multiple phases winding forwards, which isolated), coupled via individual edge-based phase-resetting mechanisms, with

appropriate time-delays. The results are the same as those for the full VLS DSS – internal, hierarchically-arranged, dynamical modes responding to external stimulation. Yet these *Kuramoto*-type simulations only require 1M or so multi dimensional clocks, with say 10M degrees of freedom in total. Whereas the full VLS DSS simulations require 10B degrees of freedom. As a result the reduced system may run on a laptop (dual core), as opposed to a supercomputer [20,32].

Over many years various *toy* circuits built with neurons have been investigated. But this is a red herring. The full scale simulations with realistic architectures and dynamics had to wait for suitable computing platforms. As a result it is clear that the possibility of VLS simulations producing a dynamical systems and network science enabled response to the hard problem was discounted prematurely. Once investigators could peer inside such systems and reverse engineer them (in a way that is impossible for human brains, given the resolution of even the most powerful scanners), the internal dynamics became apparent. The *Entwinement Hypothesis* [20] is thus a logical outgrowth of VLS DSS.

Much of the earlier philosophical work often argued that cognition and consciousness are separate, or that cognition begets consciousness as a consequence or by-product of processing (see the multiple drafts hypothesis [33], for example). However, it is now suggested that one should accept the corollary (from the insights) gained via DSS, that internal conscious phenomena are crucial to certain efficiencies within cognition. Cognition and consciousness would be thus mutually dependent, and entwined [20].

2 Comparisons

DSS considers an open dynamical system containing up to 10B neurons embedded within a directed network-of-networks that is irreducible (strongly connected) and is subject to a continuous stream of sensory inputs, yet it responds in consistent ways. It moves from causes to effects - from stimuli to decisions, inferences, and instantiating appropriate internal modes. The structures employed rest on what is observed in terms of neuronal dynamics, cortex architecture, and transmission time lags. DSS enables the analysis and reverse engineering of the integrated system behaviour, including the discovery of internal latent modes, which are hypothesised to be physical causes of sensations and qualia. DSS shows how these in turn can influence and constrain immediate cognition. These conclusions are thus based on the observed brain structure and behaviour, and on a multitude of DSS experiments.

On the other hand IIT moves in the opposite direction, It starts out from a generalised irreducible (strongly connected) and agile system, and measures the integrated (whole-system) performance via Φ . In fact Φ really seeks to measure a whole range of possible dynamical phenomenon, including all possible internal response modes to incoming stimuli. Thus, within its generality, IIT subsumes the internal responsive structures that are exhibited by particular systems, yet it does not explicitly demonstrate the existences dynamical modes within the integrated response. IIT does not rely on the specific network-of-networks architecture, only properties of it; and consequently IIT is not able make testable

predictions (such as having a fairly uniform size of neural columns [18,19] in maximising the total number of dynamical degrees of freedom). The power of having a measure lies not in its derivation (and well-definedness) but in introducing a systems-level concept beyond energy, entropy, and complexity measures (such as modularity).

Both IIT and DSS are described by similar vocabulary and they exhibit the same obvious role for evolutionary cognitive and consciousness development. Assuming that high- Φ induces some advantages to an organism, such as the preconditioning and hence fast-thinking advantage [21] implied by DSS, then the brain can have evolved in structural form and dynamics so as to increase this.

IIT and VLS DSS are really the same thing but coming from different directions. DSS constructs a **bottom-up** narrative of *what occurs within* [19] for a very specific class of cortex-like systems, making specific and testable predictions based on observed structure and experimentation. IIT provides a much more general setting, a **top-down** view, and it asserts that a high level of a suitably defined performance measure can imply the existences of conscious internal phenomena.

QC is a rather special case of a theoretical approach offering a (presently) theoretical solution. It comes with no practical justification nor evidence for its establishment and relevance, and yet it supplies some sophisticated benefits - elements that deal with uncertainty and also seek to explain why conscious phenomena are elusive and beyond physical measurement (observation).

The evolutionary question is important for QC, and quantum biology in general. Has biology evolved so as to exploit quantum effects within *warm and wet* environments, on the increasing spatial scales of molecules, cells, organs and organisms? If not why not? Does cellular and systems biology take place at the wrong scales for quantum effects to be relevant? The advantages of quantum effects within cognitive and conscious performance might be very great, if ever achievable. Objections have encouraged proponents to become more specific about where and how quantum effects might ever arise within the human brain [8], and yet still persist.

QC says nothing about relative levels of consciousness (compared to IIT) and nothing at all about the brain's evolved architecture or the plethora and role of inner sensations (compared with VLS DSS); beyond seeking sub-cellular structures that might support any quantum effects. Instead it provides a theoretical *raison d'être* for conscious experiences.

Of course DSS is classical, and far simpler and more straightforward than QC. It is also testable and produces observable consequences, including support for the *Entwinement hypothesis* [20]. Moreover, any DSS progress at all required the development of supercomputing facilities that could simulate such VLS dynamics [22,26,27]. Hence such a classical approach (as set out in [19]) was held up until about five years ago. Perhaps its efficacy was simply discounted too early by commentators; since (human) "nature abhors a vacuum".

VLS DSS implies that QC is unnecessary. QC implies that whatever DSS demonstrates is irrelevant.

Very usefully, in theory IIT applies to both classical and quantum approaches [14, 15]. Yet any implementation requires some detailed descriptions of the system architecture and dependencies of the systems' elements and mechanisms.

It would be fascinating if IIT could ever calculate Φ for the same systems set out and deployed within DSS, for both the VLS DSS and the simpler Kuramoto-style, network of multidimensional clocks systems. This would be a very good next step.

Furthermore, any physiological surrogate for Φ , possibly tied to some evolutionary advantages, would be extremely useful. We can argue that DSS shows us some facets of the dynamics and architecture (the total dynamical degrees of freedom, for example) that would confer fast-thinking advantages. We can also observe many physiological surrogates for individual inner feelings (blushing, trembling, non-poker faces, heart rate, cortisol, and so on). Could we identify some more generalised observables that might be a surrogates for the full measure, Φ ?

In summary, we suggest that the best next steps for IIT should be (i) to ground it further to the specific system observed within the cortex, from where DSS starts out; (ii) and identify appropriate physiological markers that are aligned with Φ . For VLS DSS the immediate experimental challenge is to identify evidence for the existence of specific internal dynamical modes corresponding to certain internal sensations. Such a step requires high resolution neuroimaging, over time as well as across the cortex (not highly localised), relating cognitive and consciousness entwinement more closely to the recent the progress on *neural correlates of consciousness* [34]. The reverse engineering of massive ensembles of VLS simulations creates its own "big data" problem. The methodology deployed in [19, 32] should be improved and made more transparent.

References

1. Chalmers, D.J.: Facing up to the Problem of Consciousness. *J. Conscious. Stud.* **2**, 200–19 (1995)
2. Chalmers, D.J.: *The Conscious Mind: In Search of a Fundamental Theory*. Oxford University Press, New York (1996)
3. Penrose, R.: *The Emperor's New Mind*. Penguin Books, New York (1989). ISBN 0-14-01-4534-6
4. Jedlicka, P.: Revisiting the quantum brain hypothesis: toward quantum (Neuro)biology? *Front. Mol. Neurosci.* **10**, 366 (2017). PMID: 29163041; PMCID: PMC5681944. <https://doi.org/10.3389/fnmol.2017.00366>
5. Lambert, N., Chen, Y.N., Cheng, Y.C., et al.: Quantum biology. *Nat. Phys.* **9**, 10–18 (2013). <https://doi.org/10.1038/nphys2474>
6. Tegmark, M.: Importance of quantum decoherence in brain processes. *Phys. Rev. E* **61**, 4194–4206 (2000). <https://doi.org/10.1103/PhysRevE.61.4194>
7. Seife, C.: Cold numbers unmake the quantum mind. *Science* **287**(5454), 791 (2000)
8. Penrose, R., Hameroff, S.: Consciousness in the universe: neuroscience, quantum space-time geometry and Orch OR theory. *J. Cosmol.* **14** (2011)

9. Marai, A., et al.: The future of quantum biology. *J. R. Soc. Interface* **15**(148):20180640 (2018). PMID: 30429265; PMCID: PMC6283985 <https://doi.org/10.1098/rsif.2018.0640>
10. Chalmers, D.J., McQueen, K.J.: Consciousness and the collapse of the wave function. *Consciousness Quant. Mech.* **11** (2022)
11. Kremnizer, K., Ranchin, A.: Integrated information-induced quantum collapse. *Found. Phys.* **45**(8), 889–899 (2015)
12. Tononi, G.: An information integration theory of consciousness. *BMC Neurosci.* **5**(1), 42 (2004). ISSN 1471–2202. PMC 543470. PMID 15522121. <https://doi.org/10.1186/1471-2202-5-42>
13. Oizumi, M., Albantakis, L., Tononi, G.: From the phenomenology to the mechanisms of consciousness: integrated information theory 3.0. *PLoS Comput. Biol.* **10**(5), e1003588 (2014)
14. Kleiner, J., Tull, S.: The mathematical structure of integrated information theory. *Front. Appl. Math. Stat.* **6** (2020). <https://doi.org/10.3389/fams.2020.602973>
15. Zanardi, P., Tomka, M., Venuti, L.C.: Quantum Integrated Information Theory. arXiv preprint [arXiv:1806.01421](https://arxiv.org/abs/1806.01421) (2018). Comparison with Standard Presentation of IIT 3.0
16. Lau, H.: Open letter to NIH on Neuroethics Roadmap (BRAIN initiative) 2019. In: *Consciousness We Trust* (2020). <https://inconsciousnesswetrust.blogspot.com/2020/05/open-letter-to-nih-on-neuroethics.html>
17. Merker, B.: The integrated information theory of consciousness: a case of mistaken identity. *Behav. Brain Sci.* **45**, e41 (2021)
18. Grindrod, P., Lee, T.E.: On strongly connected networks with excitable-refractory dynamics and delayed coupling. *Roy. Soc. Open Sci.* **4**(4), 160912 (2017). <https://doi.org/10.1098/rsos.160912>
19. Grindrod, P., Lester, C.: Cortex-like complex systems: what occurs within? *Front. Appl. Math. Stat.* **7**, 51 (2021). <https://www.frontiersin.org/article/10.3389/fams.2021.627236>
20. Grindrod, P., Brennan, M.: Cognition and consciousness entwined. *Brain Sci.* **13**(6), 872 (2023). <https://doi.org/10.3390/brainsci13060872>
21. Kahneman, D.: *Thinking, Fast and Slow*, Farrar, Straus and Giroux, New York (2011)
22. Furber, S.B., Galluppi, F., Temple, S., Plana, L.A.: The SpiNNaker project. *Proc. IEEE* **102**(5), 652–665 (2014)
23. Eliasmith, C., Trujillo, O.: The use and abuse of large-scale brain models. *Curr. Opin. Neurobiol.* **25**, 1–6 (2014). <https://doi.org/10.1016/j.conb.2013.09.009>
24. Chen, S., et al.: How big data and high-performance computing drive brain science. *Genomics Proteomics Bioinform.* **17**, 381–92 (2019). <https://doi.org/10.1016/j.gpb.2019.09.003>
25. DeBole, M.V., et al.: TrueNorth: accelerating from zero to 64 million neurons in 10 years. *Computer* **52**(5), 20–9 (2019)
26. Temple, S., Furber, S.: Neural systems engineering. *J. R. Soc. Interf.* **4**(13), 193–206 (2007). CrossRef Full Text — Google Scholar <https://doi.org/10.1098/rsif.2006.0177>
27. Moss, S.: SpiNNaker: Brain Simulation Project Hits One Million Cores on a Single Machine Modeling the Brain Just Got a Bit Easier. *Data Center Dynamics* (2018). <https://www.datacenterdynamics.com/en/news/spinnaker-brain-simulation-project-hits-one-million-cores-single-machine/>. Accessed 16 Oct 2018

28. Von Neumann, J.: *Mathematical Foundations of Quantum Mechanics/Mathematische Grundlagen der Quantenmechanik*. Springer (1932)
29. Wigner, E.P.: Remarks on the mind-body question. In: Mehra, J. (ed.) *Philosophical Reflections and Syntheses*, pp. 247–260. Springer, Heidelberg (1995). https://doi.org/10.1007/978-3-642-78374-6_20
30. Wikipedia. *The Human Brain Project* (2020). https://en.wikipedia.org/wiki/Human_Brain_Project(extracted May, 2020)
31. Mullin, E.: How big science failed to unlock the mysteries of the human brain, MIT Tech Review (2021). <https://www.technologyreview.com/2021/08/25/1032133/big-science-human-brain-failure/>
32. Grindrod, P., Brennan, M.: Generalised Kuramoto models with time-delayed phase-resetting for k -dimensional clock., *Brain Multiphys.* **4** (2023). <https://doi.org/10.1016/j.brain.2023.100070>
33. Dennett, D.C.: *Consciousness Explained*, Little, Brown and Co (1991)
34. Nani, A., Manuello, J., Mancuso, L., Liloia, D., Costa, T., Cauda, F.: The neural correlates of consciousness and attention: two sister processes of the brain. *Front. Neurosci.* **31**(13), 1169 (2019)

Author Index

A

Ahmadifar, Saeid 172
Allendorf, Daniel 408
Andraud, Mathieu 325
Apolloni, Andrea 325
Arsevska, Elena 325
Asgari, Yasaman 209
Ataides, Juan Pablo Bonilla 289
Aurachman, Rio 362
Ayres, Fabio 3

B

Baughman, Brian 55
Bolajoko, Muhammad-Bashir 325
Bonneault, Mélanie 312
Börger, Christoph 457
Borgnat, Pierre 209
Böttcher, Philipp C. 95
Brissette, Christopher 261
Buchanan, Ashley L. 350

C

Caines, Peter E. 395
Cardinale, Eric 325
Cazabet, Remy 209
Chacko, George 3
Chen, Guanqing 15
Cherifi, Chantal 235, 248, 273
Cherifi, Hocine 235, 248, 273
Christopoulos, Konstantinos 68
Ciss, Mamadou 325
Conroy, John M. 55
Couthures, Anthony 383
Czaja, Wojciech 108

D

Dabke, Devavrat Vivek 445
De Smedt, Johannes 146
Delaplace, Franck 223
Delarocque-Astagneau, Elisabeth 312
Diallo, Cherif 235, 248, 273
Diallo, Diaoulé 338

Diop, Issa Moussa 235, 248, 273
Dorabiala, Olga 445
Dragovic, Natasa 457
Draždilová, Pavla 159
Dunyak, Alex 395

E

Eubank, Stephen 43

F

Flauder, Maxime 312
Frédéric, Guinand 435
Friedman, Samuel R. 350

G

Galante, Amanda 108
Gardner, Shari 121
Gh. Ghahramani, S. AmirAli 172
Gilmour, Samuel 470
Gnecco, Giorgio 146
Gomez, Rod 55
Gregnanin, Marco 146
Grindrod, Peter 492
Gunjača, Iva 421
Gupta, Bineet Kumar 373
Gupta, Sanjay 373

H

Haensch, Anna 457
Hecking, Tobias 338
Hemmatyar, Ali Mohammad Afshin 172
Hong, Seok-Hee 289
Hosseinioun, Morteza 172

I

Ijioma, Sandra 325

K

Kaiser, Franz 95
Kaliszewski, Ryan 55
Kamiński, Bogumił 133
Katenka, Natallia V. 350

Kelley, Cassandra 121
 Kirshtein, Arkadz 457
 Kogut, Stephen 350
 Kok, Rowena 289
 Korobskiy, Dmitriy 3
 Kudelka, Milos 197

L

La, Richard J. 300
 Lasaulce, Samson 383
 Latora, Vito 95
 Lemire, Gabrielle 350
 Lines, Nicholas A. 55
 Liu, Baqiao 3
 Lyu, Wenruo 82

M

Mallavarapu, Aditi 121
 Meidiana, Amyra 289
 Mesdour, Asma 325
 Mirkin, Boris 185
 Molino, Neil 55
 Monod, Anthea 29
 Morărescu, Irinel Constantin 383
 Mroué, Bouchra 383

O

O'Malley, A. James 15
 Ochodkova, Eliska 197
 Opatowski, Lulla 312

P

Pailodi, Vidya Kamath 3
 Pandey, Ujwal 261
 Papik, Lukas 197
 Park, Kunsoo 289
 Park, Minhyuk 3
 Parton, Maurizio 146
 Platoš, Jan 159
 Prałat, Paweł 133
 Prokop, Petr 159
 Punzo, Giuliano 362

R

Ramachandran, Rajiv 3
 Ramavarapu, Vikram 3

Ramrakhiyani, Madhvi 484
 Ronellenfitsch, Henrik 95
 Roschelle, Jeremy 121

S

Samhan, Natalie 421
 Samiee, Hojjat 172
 Savkli, Cetin 108
 Schönfeld, Jurij 338
 Schwartz, Catherine 108
 Sefer, Emre 470
 Shalileh, Soroosh 185
 Shimada, Ayako 350
 Skov, Benjamin 350
 Slota, George M. 261
 Srivastava, Rishi 373
 Sunitha, V. 484

T

Tabatabaee, Yasamin 3
 Théberge, François 133
 Thiébaud, Anne C. M. 312
 Tiwari, Mukesh 484
 Treur, Jan 421
 Tsichlas, Konstantinos 68

U

Uzzo, Stephen 121

V

Verma, Satya Bhushan 373
 Vincent, Bridonneau 435

W

Walker, Erin 121
 Warnow, Tandy 3
 Witthaut, Dirk 95
 Wolf, Arne 29

Y

Yadav, Abhay Kumar 373
 Yoann, Pigné 435

Z

Zajac, Sebastian 133
 Zhao, Liang 82